

School of Information Systems, Curtin Business School

**Copyright Protection of Scalar and Multimedia Sensor Network
Data Using Digital Watermarking**

Bambang Harjito

**This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University**

August 2013

DECLARATION

To the best of my knowledge and belief this thesis contains no material previously published by any other person, except where due acknowledgement has been made.

Further, this thesis contains no material which has been accepted for the award of any other degree or diploma by any university.

Signed: _____

Date: _____

ABSTRACT

Wireless Sensor Networks (WSNs) are capable of sensing, processing and carrying out wireless communication, all through a tiny embedded device. Now, the availability of low-cost cameras, along with complementary metallic oxide semiconductor (CMOS) image sensors and microphones, that have a wide range of ubiquitous applications in capturing multimedia content from different environments, has led to the development of Wireless Multimedia Sensor Networks (WMSNs), i.e., networks of interconnected, wireless devices that allow the retrieval of not just scalar sensor data and still images but also audio and video streams from the environment. However, since information in WMSNs is multimedia in nature, and uses a wireless link as the mode of communication, this poses a serious security threat to WMSNs, making them vulnerable to different types of intentional network attacks, e.g., eavesdropping, modification, routing attack, camouflages adversaries and man-in-the-middle attack. Besides, WMSNs also suffer from bad stem as some sensor nodes have video cameras along with high computation abilities, due to which the authenticity of the data transmitted cannot be ensured. Further, man-in-the middle attack can cause modification in the transmitted data (alter or delete data, or insert extraneous or false data) while a bad network channel may introduce noise into the signal, causing data damage. It is important to address these challenges to make WMSNs secure and trustworthy. Watermarking techniques are among the ways currently being investigated to address some of these challenges, like broadcast monitoring, copyright protection, tampering and ownership attacks, as attractive alternatives to traditional techniques, because of their light resource requirements. A watermark adds a second line of defence to ensure that the data is valid, even if someone cracks the encryption.

This research investigates different watermarking techniques to address the issue of copyright protection of the scalar data in WSNs and image data in WMSNs, in order to ensure that the proprietary information remains safe between the sensor nodes in both. The first objective of the research was to develop a Linear Feedback Shift Register (LFSR) and Kolmogorov Rule (LKR) watermarking technique for the copyright protection of scalar data in WSNs. The LKR watermarking technique developed can protect the copyright data from deletion, packet replication and multiple data identities (data Sybil attack), although it is ineffective against false data insertion, data modification and selective forwarding. The second objective of the research was to develop a Gaussian Pyramid Kolmogorov Rule (GPKR) watermarking technique for copyright protection of image data in WMSN. The GPKR watermarking technique developed can protect the copyright image data in WMSNs from insertion and replication, although it cannot protect them from deletion and modification.

ACKNOWLEDGEMENTS

I take great pleasure in thanking all those who have helped me in successfully completing this dissertation. In particular, I express my deepest gratitude to my supervisors, Dr. Vidyasagar Potdar and Dr Song Han, who continually encouraged and guided me during all these years of strenuous research. Their own enthusiasm, inspiration and encouragement, patient effort in explain things clearly and simply, sound advice and teaching, elevating company and numerous ideas, have been of invaluable help to me. I would simply have been lost without them.

I also take this opportunity to express my deepest gratitude to my co-supervisor, Dr. Jaipal Singh, for his invaluable critiques, insights and suggestions that went a long way in improving the research and the thesis.

My specials thanks are due to Prof Elizabeth Chang for her continued assistance during the whole period of my study. Her constant guidance and encouragement throughout my PhD program were of really great help. I do pray for her to be always blessed with joy and happiness. My special thanks also extended to thank to Prof Kate Wright and Peter Dell for their supporting and helping the whole periode of my study.

Many of my colleagues have obliged me with their peer reviews and helpful suggestions. In particular, I would like to mention Dr. Hai Dong, and my best friends, Zia Ur Rahman, Sazia, and Dinusha.

I am heavily indebted and deeply thankful to my wife, Rini Prasetyowati, and my two sons, Miftah Nafi Hisyam and Faishal Tsaqib Khoiry, who kept their faith in me, and allowed my ambitions to soar as high as they could. It were their watchful eyes and loving encouragement, whenever they perceived any slackness, that kept me motivated, committed, and able to tackle challenges head on.

Finally, I would like to thank the Government of Indonesia and the Directorate of Higher Education (DIKTI) for their financial support during the entire period of my study; my institution, Sebelas Maret University; and all the people of Indonesia; whose love inspired me to undertake this research.

TABLE OF CONTENTS

DECLARATION	II
ABSTRACT.....	III
ACKNOWLEDGEMENTS	IV
TABLE OF CONTENTS	V
LIST OF FIGURES.....	XI
LIST OF TABLES	XV
LIST OF ALGORITHMS	XVII
LIST OF ABBREVIATION AND CRONYMS	XVIII
LIST APPENDICES	XIX
CHAPTER ONE	
INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Wireless Sensor Network – An Overview	2
1.2.1 WSNs Architecture	3
1.2.2 Sensor Node Hardware Architecture.....	4
1.2.3 Sink and Gateway Nodes	5
1.2.4 Sensor Networks Management.....	6
1.2.5 WSN Applications.....	7
1.3 Wireless Multimedia Sensor Networks - An Overview	13
1.3.1 WMSN architecture.....	13
1.3.2 Multimedia Sensor Node Hardware Architecture	14
1.3.3 WMSNs Applications	17
1.4 Digital Watermarking – An Overview	20
1.4.1 Digital Watermarking Applications	22
1.4.2 Digital Watermarking Process.....	24
1.5 Digital Watermarking Techniques for WSNs	29
1.6 Digital Watermarking Technique for WMSNs	31
1.7 Research Motivations.....	33
1.7.1 Copyright Protection of Scalar data	33
1.7.2 Copyright Protection of Images	34
1.8 Research Objectives	34
1.8.1 Developing Digital Copyright Protection of Scalar Data in WSNs using Watermarking Technique.....	35
1.8.2 Developing Digital Copyright Protection of Multimedia Data in WMSNs using Watermarking Technique.....	35

1.9 Significance of the Research.....	36
1.9.1 Social Significance.....	36
1.9.2 Economic Significance.....	37
1.9.3 Scientific Significance.....	37
1.10 Structure of the Dissertation.....	38
1.11 Conclusion.....	39
CHAPTER TWO	
LITERATURE REVIEW.....	40
2.1 Introduction.....	40
2.2 Theoretical Background.....	41
2.2.1 Atomic Trilateration.....	41
2.2.2 Linear Feedback Shift Register (LFSR).....	44
2.2.3 Kolmogorov Complexity Rule.....	45
2.2.4 The Gaussian Pyramids.....	46
2.3 Security in WSNs and WMSNs.....	49
2.3.1 Security Requirements in WSNs and WMSNs.....	49
2.3.2 Types of Attacks in WSNs and WMSNs.....	53
2.4 Digital Watermarking Technique.....	60
2.4.1 Classification of Watermarks.....	61
2.4.2 Watermark Embedding Strategies.....	63
2.4.3 Watermark Extraction Strategies.....	65
2.4.4 Watermark Detecting Strategies.....	66
2.4.5 Security Requirements for Digital Watermarking Technique.....	67
2.4.6 Attacks on Digital Watermarking Techniques.....	69
2.4.7 Working of Digital Watermarking Technique in WSNs.....	72
2.4.8 Working of Digital Watermarking Technique in WMSNs.....	73
2.5 Evaluation Framework for Digital Watermarking Techniques in WSNs.....	75
2.6 Survey of Literature on Digital Watermarking Technique in WSNs.....	76
2.6.1 Cover Medium.....	77
2.6.2 Watermark Message.....	78
2.6.3 Sensed Data.....	80
2.6.4 Type of Watermarks.....	82
2.6.5 Watermark Key.....	83
2.6.6 Watermark Generator.....	85
2.6.7 Watermark Embedding Technique.....	87
2.6.8 Watermark Detecting Technique.....	89
2.6.9 Noise.....	90

2.6.10 Vulnerability Attacks	91
2.7 Survey of Literature on Digital Watermarking Technique in WMSNs	93
2.7.1 Cover Medium.....	94
2.7.2 Sensed Data	96
2.7.3 Type of Watermarks	97
2.7.4 Watermark Key	99
2.7.5 Transform Domain	101
2.7.6 Watermark Generator	101
2.7.7 Watermark Embedding Technique.....	102
2.7.8 Watermark Detection Technique.....	105
2.7.9 Noise	106
2.7.10 Vulnerability Attacks	107
2.8 Summary of Issues in Digital Watermarking for WSNs and WMSNs	108
2.9 Conclusion.....	113
CHAPTER THREE	
PROBLEM DEFINITION	114
3.1 Introduction	114
3.2 Problem definition.....	115
3.2.1 Problems with copyright protection of scalar data in WSNs	116
3.2.2 Problem with copyright protection of images in WMSNs	118
3.3 Research Issues	121
3.3.1 Research Issue 1: Developing Copyright Protection of Scalar data in WSNs using Watermarking Technique.....	121
3.3.2 Research Issue 2: Developing Copyright Protection of Images in WMSNs using Watermarking Technique.....	122
3.4 Research Methodology.....	123
3.4.1 Problem definition.....	123
3.4.2 Conceptual solution.....	124
3.4.3 Implementation, test and evaluation.....	124
3.5 The Objective of the Research	124
3.6 Summary the problem Definition & Research Issues	125
3.7 Conclusion.....	126
CHAPTER FOUR	
AN OVERVIEW OF THE SOLUTION AND THE CONCEPTUAL PROCESS	127
4.1 Introduction	127
4.2 The Proposed Solution : Overview	127
4.3 Solution Description.....	129

4.3.1 LFSR and Kolmogorov Rule for Copyright Protection of Scalar Data in WSNs (LKR Watermarking Technique).....	130
4.3.2 Gaussian Pyramids and Kolmogorov Rule for Copyright Protection of Images in WMSNs (GPKR Watermarking Technique).....	131
4.4 Conceptual process.....	133
4.4.1 Requirements, Elicitation and Prioritization	134
4.4.2 Design rationale.....	134
4.4.3 Theoretical foundation	134
4.4.4 Prototype Implementation	135
4.4.5 Experimental Setting.....	135
4.4.1 Results and Observations	135
4.4.2 Validation and Comparative Analysis.....	136
4.5 Conclusion.....	136
CHAPTER FIVE	
LKR WATERMARKING TECHNIQUE.....	137
5.1 Introduction	137
5.2 Proposed LKR Watermarking Technique	138
5.2.1 General overview of LKR	138
5.2.2 Requirements.....	138
5.2.3 Design Rationale	139
5.3 Theoretical Foundation for LKR Watermarking Technique	141
5.3.1 Cover Medium Generation.....	143
5.3.2 Watermark Generation	145
5.3.3 Watermark Embedding Algorithm.....	152
5.3.4 Watermark Extraction & Detection Algorithm	154
5.4 Implementation of the Prototype.....	159
5.4.1 Source Code : Network Set-Up Generation	160
5.4.2 Source Code: Cover Medium Generation	160
5.4.3 Source Code : Watermark Generation	161
5.4.4 Source Code : Embedding Watermark Signal.....	161
5.4.5 Source Code: Extracting Watermark.....	162
5.4.6 Source Code: Attacks	163
5.5 Experimental Setting	164
5.5.1 Network Set -Up.....	164
5.5.2 Parameters	173
5.5.3 Attacks Characterization	174
5.6 Results and Observations	179
5.7 Validation and discussion.....	196

5.8 Comparative analysis	198
5.9 Conclusion.....	200
CHAPTER SIX	
GPKR WATERMARKING TECHNIQUE	201
6.1 Introduction	201
6.2 The Proposed GPKR Watermarking Technique	201
6.2.1 A General Overview of GPKR Watermarking Technique.....	202
6.2.2 Requirements.....	202
6.2.3 Design Rationale	203
6.3 Theoretical Foundation for GPKR Watermarking Technique.	205
6.3.1 Cover Medium Generation.....	206
6.3.2 Watermark Generation	209
6.3.3 Watermark Embedding Algorithm.....	216
6.3.4 The Process of Extraction and Detection of the Watermark	218
6.4 Implementation of the Prototype.....	224
6.4.1 Source Code : Network Set -Up Generation	226
6.4.2 Source Code : Cover Medium Generation	226
6.4.3 Source Code : Watermark Generation.....	226
6.4.4 Source Code : Embedding Watermark	228
6.4.5 Source Code : Extracting and Detecting	228
6.5 Experimental Setting	230
6.5.1 Network Set-Up.....	231
6.5.2 Parameters	243
6.5.3 Attack Characterization.....	244
6.6 Results and Observations	250
6.7 Validation and Discussion.....	259
6.8 Comparative Analysis	261
6.9 Conclusion.....	262
CHAPTER SEVEN	
CONCLUSION AND FUTURE WORK.....	263
7.1 Introduction	263
7.2 Problems and Issues	264
7.2.1 Problems with Copyright Protection of Scalar Data in WSNs.....	264
7.2.2 Problems with Copyright Protection of Images in WMSNs	265
7.3 Contributions of the Thesis	265
7.4 Future Works.....	268
BIBLIOGRAPHY	270

APPENDIX I.....	280
APPENDIX II.....	297

LIST OF FIGURES

Figure 1.1 Wireless Sensor Networks Architecture	3
Figure 1.2 General hardware architecture of a sensor node (Akyildiz, Melodia and Chowdhury, 2007).....	4
Figure 1.3 Network management architecture of WSN.....	6
Figure 1.4 How drones work (BBC and Asia 2012)	9
Figure 1.5 a. Repackaging sensor node b. Placement of nodes within a tree (Gilman, Joseph and Kevin, 2005)	10
Figure 1.6 Sensor network for flood detection (Elizabeth, Sai, and Daniela 2008).....	12
Figure 1.7 Wireless Multimedia Sensor Networks architecture.....	14
Figure 1.8 General Hardware architecture of multimedia sensor node (Akyildiz, Melodia, and Chowdhury 2008).....	15
Figure 1.9 (a) SparkFun CMUcam4 (Xiao, 2006) b. Stargate (Akan, Pascal Zhang and Qian Jayant, 2008).....	17
Figure 1.10 Watermark embedding process.....	21
Figure 1.11 Watermark detection process.....	22
Figure 1.12 Key components digital watermarking system.....	25
Figure 1.13 An example of the use of watermark in WSNs.....	30
Figure 1.14 An Example of the use of watermark in WMSNs	32
Figure 2.1 Atomic trilateration process.....	41
Figure 2.2 Block of Linear Feed Back Shift Register	45
Figure 2.3 A graphical representation of the process in one dimension	47
Figure 2.4 An example of reducing an image	48
Figure 2.5 An example of expanding the reduced image to the size of the original	49
Figure 2.6 General classification of security attacks (Lingxuan and Evans 2003)	54
Figure 2.7 Classification of security attacks on WSNs and WMSNs (Lingxuan and Evans 2003)	55
Figure 2.8 Classification of digital watermarks	62
Figure 2.9 NLSP used as the cover medium	64
Figure 2.10 Watermark constraints	64
Figure 2.11 Watermark constraints added to the cover medium.....	64
Figure 2.12 A broad classification of currently known attacks on digital watermarking techniques	70
Figure 2.13 A perceptual remodulating attack	71
Figure 2.14 Working of digital watermarking technique in WSNs.....	73
Figure 2.15 Working of digital watermarking technique in WMSNs	74
Figure 2.16 The 10 parameters for evaluation of digital watermarking in WSNs	76
Figure 2.17 Cover medium.....	77

Figure 2.18 Watermark message.....	79
Figure 2.19 Encoded watermark message.....	79
Figure 2.20 An example of sensed data encoded by hash function.....	81
Figure 2.21 Types of watermarks used in WSN	82
Figure 2.22 Type of watermark keys	84
Figure 2.23 The 10 parameters for the evaluation of digital watermarking techniques in WMSNs	94
Figure 2.24 Cover media used in WMSN literature.....	95
Figure 2.25 Types of watermarks for WMSNs	98
Figure 2.26 Types of watermark keys used in WMSNs.....	100
Figure 2.27 (a) The two adaptive threshold (Honggang et al. 2008), (b) the weight coefficient of the watermark in DCT (Pingping, Yao Jiangtao, and Zhang Ye 2009) and (c) Orthogonal Frequency Davison Multiplexing (OFDM) in FFT (Masood, Haider, and Sadiq-ur 2010).....	104
Figure 3.1 Copyright protection of scalar data between sensor nodes in WSNs.....	115
Figure 3.2 Copyright protection of images between multimedia sensor nodes in WMSNs	115
Figure 3.3 Copyright protection of scalar data in WSNs	117
Figure 3.4 The architecture of CMOS image sensor system performing in the transform.	119
Figure 4.1 An overview of the conceptual solution	129
Figure 4.2 Working LKR Watermarking technique for copyright protection of scalar data in WSNs	131
Figure 4.3 Working GPKR Watermarking technique for copyright protection of images in WMSNs	132
Figure 4.4 Conceptual process followed in this thesis	133
Figure 5.1 Theoretical foundation of LKR watermarking technique in WSNs	140
Figure 5.2 The process of the creation of message sensed data	141
Figure 5.3 Flowchart of the generation of Cover Medium.....	145
Figure 5.4 Flowchart for conversion of decimal data into binary digits	147
Figure 5.5 Flowchart for generation of watermark signal.....	149
Figure 5.6 Flowchart for the generation of watermark constraints	150
Figure 5.7 Flowchart for the generation of message sensed data.....	151
Figure 5.8 Flowchart for watermark embedding algorithm	154
Figure 5.9 Flowchart for extraction of message sensed data	156
Figure 5.10 Flowchart for the process of detection of watermark signal.....	158
Figure 5.11 Screenshot of the cover medium generation process using MATLAB code	160
Figure 5.12 Screenshot of the process of generating watermark signal and watermark constraints using MATLAB code	161
Figure 5.13 Screenshot of the process of embedding watermark signal using MATLAB code	162
Figure 5.14 Screenshot of the process of extracting watermark signal using MATLAB code	163

Figure 5.15 Flowchart for setting up LKR watermarking technique	166
Figure 5.16 Screenshot of the network setting for LKR Watermarking Technique using MATLAB Code.....	167
Figure 5.17 The 75 nodes randomly deployed within a 500 meter square area	167
Figure 5.18 Screenshot of generating watermark signal MATLAB Code.....	171
Figure 5.19 Screenshot of generating and extracting the signal using MATLAB Code.....	173
Figure 5.20 The categorization of attacks that affect communication.	175
Figure 5.21 Insertion of false watermark constraints into the cover medium	176
Figure 5.22 Deletion of watermark constraint data from the cover medium	177
Figure 5.23 Watermark constraints data replication attack in the cover medium	178
Figure 5.24 Watermark constraints in the cover medium in Sybil data attack.....	179
Figure 5.25 Screenshot of watermark embedding process Matlab Code	180
Figure 5.26 The value of error in the cover medium process.....	182
Figure 5.27 The value of error in watermark constraints embedding process	184
Figure 5.28 The value of error for false data attack in WSNs.....	186
Figure 5.29 The value of error data deletion attack in WSNs	189
Figure 5.30 The value of error data replication attack in WSNs.....	192
Figure 5.31 The value of error of Sybil data attack in WSNs	195
Figure 5.32 All watermark constraint attacks	197
Figure 6.1 The general model of GPKR watermarking technique for copyright protection of images in WMSNs.....	204
Figure 6.2 Flowchart for generation of cover medium in GPKR watermarking technique	209
Figure 6.3 Flowchart for getting the reduced image using the Gaussian pyramids	212
Figure 6.4 Flowchart for converting the reduced image to binary matrix.....	213
Figure 6.5 Flowchart for generating watermark constraints	216
Figure 6.6 Watermark constraints and digital message image embedding process	218
Figure 6.7 Flowchart for extraction process.....	221
Figure 6.8 Flow chart for the detection process of the GPKR watermarking technique.....	224
Figure 6.9 Screenshot of cover medium generation using MATLAB Code.....	226
Figure 6.10 Screenshot of cover medium generation using MATLAB Code.....	227
Figure 6.11 Screenshot of the process of embedding watermark constraints	228
Figure 6.12 Screenshot of the process of extracting the reduced image	229
Figure 6.13 Screenshot of the process of detecting the watermark constraints.....	229
Figure 6.14 Flowchart for network set-up of GPKR watermarking technique	233
Figure 6.15 Screenshot of network setting for GPKR watermarking technique using MATLAB Code.....	234
Figure 6.16 50 nodes randomly deployed within a 200 meter length and 100 meter width	234
Figure 6.17 The Lena figure (http://www.cs.cmu.edu/~chuck/lennapg/lenna.shtml).....	238

Figure 6.18 The process of reducing the Lena image	239
Figure 6.19 The Lena image expansion process	242
Figure 6.20 The value of error in watermark constraint embedding process	255
Figure 6.21 Value of the error of data deletion attack.....	256
Figure 6.22 Value of the error of false data insertion attack.....	257
Figure 6.23 Value of the error of data modification attack	258
Figure 6.24 Value of the error of replication attack	259
Figure 6.25 All watermark constraint attacks	260

LIST OF TABLES

Table 2.1 Various types of attacks in WSN and WMSNs.....	58
Table 2.2 Classification of digital watermarking techniques	60
Table 2.3 Cover media used in literature	78
Table 2.4 Watermark messages used in literature.....	80
Table 2.5 Sensed data used in literature	81
Table 2.6 Types of watermark used in literature.....	83
Table 2.7 Watermark keys used in literature.....	84
Table 2.8 Watermark generators used in literature	86
Table 2.9 Watermark embedding techniques used in literature	88
Table 2.10 Watermarking detection techniques used in literature	90
Table 2.11 Different types of noise in literature	91
Table 2.12 Attack used in literature	92
Table 2.13 Cover media used in literature	95
Table 2.14 Sensed data used in literature.....	97
Table 2.15 Types of watermark data used in literature	99
Table 2.16 The watermark keys used in literature	100
Table 2.17 Transform domains used in literature.....	101
Table 2.18 Watermark generators used in literature	102
Table 2.19 Watermarking embedding technique used in literature.....	104
Table 2.20 Watermark detecting technique used in literature.....	106
Table 2.21 Different types of noise used in literature	107
Table 2.22 Different types of attack used in literature	108
Table 2.23 Comparison of different issues in secure communication between sensor nodes for WSNs	108
Table 5.1 The particular of the Kolmogorov rule	149
Table 5.2 Coordinate positions of 75 nodes	168
Table 5.3 32 experiments of the positions of $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ and 32 experiments of time measurement using Gaussian distribution [0,1], 32 experiments of temperature randomly between [0,75] and the values of $\tau_1, \tau_2, \tau_3,$ and τ_4 generated using normal distribution [0,1].	171
Table 5.4 Parameter and its associated values used in the LKR watermarking technique.....	174
Table 5.5 The results of error for message sensed data in WSN.....	182
Table 5.6 The results of error for message sensed data and watermark constraints in WSNs	184
Table 5.7 The results of error for false data attack in WSNs	186
Table 5.8 The results of error for data deletion attack in WSNs.....	189
Table 5.9 The results of the error data replication attack in WSNs	192

Table 5.10 The results of error of Sybil data attack in WSNs.....	195
Table 5.11 A comparative analysis with other approaches copyright protection in WSNs	199
Table .6.1 The expanded Kolmogorov rule.....	214
Table 6.2 Coordinate positions of 50 nodes	235
Table 6.3 32 experiments of the positions of $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ and 32 experiments time measurement by using Gaussian distribution $[0,1]$, 32 experiment of temperature random between $[0,50]$ and the values of $\tau_1, \tau_2, \tau_3,$ and τ_4 generated by using normal distribution $[0,1]$	238
Table 6.4 Parameters and their associated values used in the GPKR watermarking technique....	243
Table 6.5 Results of data deletion attack.....	244
Table 6.6 Results of false data insertion attack.....	245
Table 6.7 New results of false data insertion	246
Table 6.8 Results of data modification attack	247
Table 6.9 Results of new data modification attack	248
Table 6.10 Results of data replication attack	248
Table 6.11 Results of new data replication	249
Table 6.12 The three positions, exact time and temperature.....	250
Table 6.13 Results of cover medium generation process.....	251
Table 6.14 The three positions, exact time and temperature, and feasibility of the value	252
Table 6.15 Results of watermark constraint embedding process	254
Table 6.16 A comparative analysis with other approaches to copyright protection in WMSNs ..	262

LIST OF ALGORITHMS

Pseudo Code 5.1	Generating cover medium	143
Pseudo Code 5.2	Converting sensitive data into binary sequences.....	146
Pseudo Code 5.3	Generating watermark signal.....	148
Pseudo Code 5.4	Generating watermark constraints.....	150
Pseudo Code 5.5	Generating message sensed data	151
Pseudo Code 5.6	Embedding watermark constraints	152
Pseudo Code 5.7	The process of extracting watermark signal.....	155
Pseudo Code 5.8	The process of detecting watermark signal	157
Pseudo Code 5.9	Network Set-up for LKR Watermarking Technique	164
Pseudo Code 6.1	Generating cover medium	207
Pseudo Code 6.2	Reduction of image by Gaussian Pyramid Transforms.....	210
Pseudo Code 6.3	Converting the reduced image to binary	212
Pseudo Code 6.4	Generating watermark constraints.....	214
Pseudo Code 6.5	Embedding watermark constraints	216
Pseudo Code 6.6	Expanding the reduced image by the pyramid transforms to get the sensory image	219
Pseudo Code 6.7	The process of detecting watermark constraints	222
Pseudo Code 6.8	Network set-up for GPKR watermarking technique	231

LIST OF ABBREVIATION AND CRONYMS

ICT	Information Communication and Technology	GPS	Global Positioning System
SN	Sensor Node	DoS	Denial of Service
WSNs	Wireless Sensor Networks	OSI	Open System International
ADC	Analogue to Digital Converter	NC	Normalized Correlation
UAVs	Unmanned Aerial Vehicles	SM	similarity measurements
AUV	Autonomous Underwater Vehicle	IDWT	Invers Discrete Wavelet Transform
WSSNs	Wireless Scalar Sensor Networks	LSB	Least Significant Bits
CMOS	Complementary Metallic Oxide Semiconductor	MSB	Most Significant Bits
WMSNs	Wireless Multimedia Sensor Nodes	DSSS	Direct Sequence Spread Spectrum
RSA	Rivest Shamir Adleman	MAC	Media Control Access
MD5	Message Digest 5	MSE	Mean Squared Error
RC4	Rivest Chiper 4	PSNR	Peak Signal Noise Ratio
SEC	State Electricity Company	IFFT	Invers Fast Fourier Transform
MCB	Mini Circuit Breaker	IDCT	Invers Discrete Cosines Transform
CCTV	Closed Circuit Television	DFT	Discrete Fourier Transform
PGP	Pretty Good Privacy	FFT	Fast Fourier Transform
JPEG	Joint Photographic Experts Group	OFDM	Orthogonal Frequency Davison Multiplexing
DCT	Discrete Cosines Transform		
DWT	Discrete Wavelet Transform		
LFSR	Linear Feedback Shift Register		
KR	Kolmogorov Rule		
LKR	Linear FSR Kolmogorov Rule		
MSD	Message Sensed Data		
GPT	Gauss Pyramid Transform		
GPKR	Gauss Pyramids Kolmogorov Rule		
NLSP	Non Linear System Programming		
MATLAB	Matrix Laboratory		
TOMLAB			
TDoA	Time Differences of Arrival		
RF	Radio Frequency		

LIST APPENDICES

APPENDIX I.....	280
APPENDIX I A : LKR Watermarking Technique Matlab Code	280
APPENDIX I B : GPKR Watermarking Technique Matlab Code.....	290
APPENDIX II	297
List Publication	297

CHAPTER ONE

INTRODUCTION

This chapter:

- ▶ introduces Wireless Sensor Networks,
- ▶ introduces Wireless Multimedia Sensor Networks,
- ▶ introduces Digital Watermarking,
- ▶ explains the significance and importance of Digital Watermarking for WSNs and WMSNs,
- ▶ explains the motivation and objectives of the research,
- ▶ outlines the structure of the dissertation.

1.1 Introduction

The rapid growth in information technology, and computer networks represented by the Internet, has brought about great changes in the lives of the people today. Moreover, the new developments in micro-electronic mechanical technology, computing technology and wireless communication technology have given rise to a whole new generation of large sensor networks with tiny sized sensor nodes that can integrate data collection, processing, wireless communication and much more (Raghu, Jayachandran Haiyun and Luo Tarek, 2006). Wireless Sensor Networks (WSNs) are deployed in the test area in the form of a large number of huge sensors nodes connected through wireless communication to form a multi-hop network of a self-organizing system (Yick, Mukherjee and Ghosal, 2008). These sensor nodes collaborate with each other to perceive, acquire and process the perceivable data and information in the network coverage area, and send these to the users (Pister 2003). A sensor network has three elements, i.e., sensors, the process of sensing the object, and the observer. If the Internet can be called a world of logically arranged information that has revolutionized the way people communicate with each other, a wireless sensor network can be called a world of logically arranged information that has revolutionized the way people communicate with the physical world.

Having introduced the sensor technology and pointed out its basic function, the next section gives an overview of the Wireless Sensor Networks (WSNs).

1.2 Wireless Sensor Network – An Overview

During the last few years, computing and communication technology have taken a quantum leap owing to the vastly improved miniaturisation (Pister, 2003). This development has made it possible to fabricate small electronic components that can sense, gather, process and communicate data to other nodes (Pister 2003). These small electronic components, called sensor nodes (SN), can sense and gather many types of data from different environments, including temperature, humidity, light intensity, wind velocity and vibrations, etc. Moreover, the wireless nature of communication makes it possible for these nodes to exchange information with other nodes or the outside world, without any physical connection with other such devices. WSNs provide economically viable solutions for a diverse range of monitoring and tracking applications, which is making them increasingly popular as data gathering, monitoring and tracking are crucial to the success of many critical missions and applications in the modern times (Yick, Mukherjee and Ghosal, 2008). Since WSNs can sense, process and communicate data wirelessly through just a tiny embedded component, they have increasingly drawn the interest of the research community in the recent past. This interest has been driven by their potential in tackling the practical and theoretical problems with embedded operating systems, wireless communication networks and network protocols, and distributed signal processing.

WSNs are primarily meant to collect and distribute critical data regarding various physical phenomena in the targeted area. They are composed of hundreds, or even thousands, of low-powered sensor nodes deployed at strategic locations, at low costs. These sensor nodes are capable of sensing, gathering and measuring different types of environmental data from the targeted area, such as temperature, pressure, humidity, sounds, vibrations, motions and pollutants. Moreover, they are capable of collaborating with other sensor nodes deployed in the area to tally their data and pass them on through the network to the main processing location. They can also transmit the sensed and gathered data directly to the user (Yick, Mukherjee and Ghosal, 2008; Potdar, Sharif and Chang, 2009). Modern WSNs are bi-directional, allowing the sensor activity to be controlled from the main processing location. Wireless sensor networks were primarily developed for military

applications, e.g. battlefield surveillance (Tatiana Bokareva, 2006; Durresti, 2008). However, today they have found use in many consumer as well as industrial applications, e.g., machine health monitoring, industrial process monitoring and control, and so on.

After this overview of WSNs, the next section moves on to WSN architecture.

1.2.1 WSNs Architecture

A typical WSN architecture can be seen in Figure 1.1. A common WSN architecture consists of many randomly deployed sensor nodes over the targeted field to measure environmental phenomena, such as temperature, pressure, humidity, motions, or even dissolved oxygen (Wen-bo, Hai-feng and Pei-gen, 2010). The sensor network system consists of sensor nodes, a sink and a sensor network management system (Ruiz, Nogueira and Loureiro, 2003; Wen-bo, Hai-feng and Pei-gen, 2010). Data from the nodes are processed by hopping via a number of routes, transferred to the sink, and finally put up on the Internet. Users can not only manage the sensor network node configuration but also monitor and publish the data

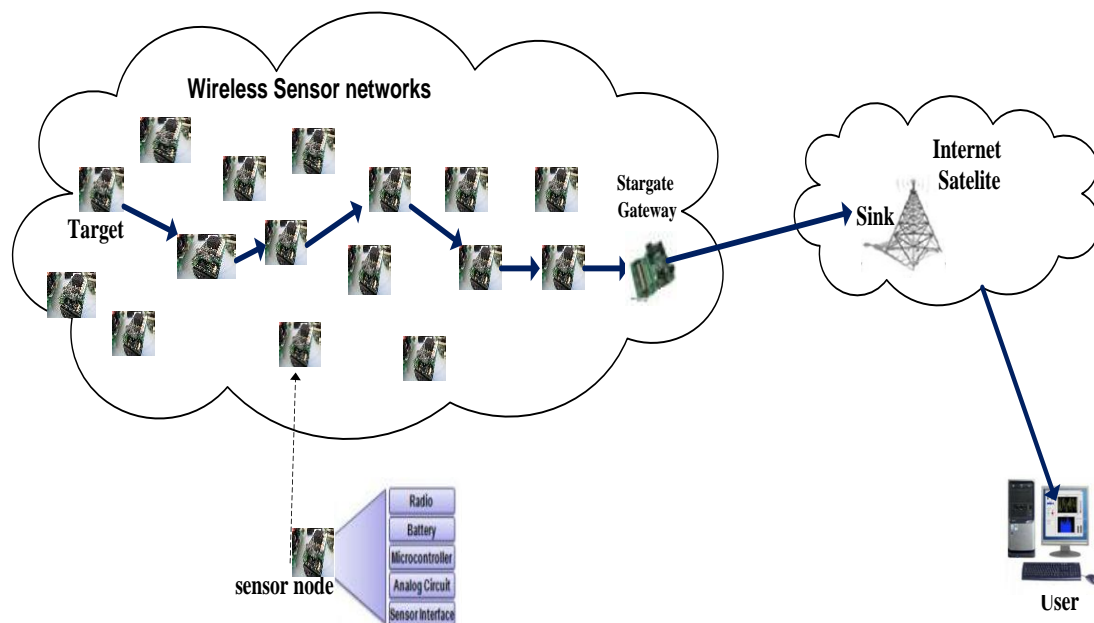


Figure 1.1 Wireless Sensor Networks Architecture

After the description of this generic WSN architecture, the next section explains the sensor node hardware architecture.

1.2.2 Sensor Node Hardware Architecture

A sensor node, which can also be referred to as a mote, is a node in a WSN that gathers and processes sensory information, and communicates with other nodes in the network (Yick, Mukherjee and Ghosal, 2008). Thus, a sensor node is a component of a larger network of sensors that collects data from the environment where it is deployed, and sends it to the main processing location of the network (Ruiz, Nogueira and Loureiro, 2003).

Figure 1.2 shows the architecture of a wireless sensor node, along with its major components, which are: sensing unit, processing unit, communication unit, and power unit. Additional units can be incorporated in a sensor node, depending on the application it is meant for.

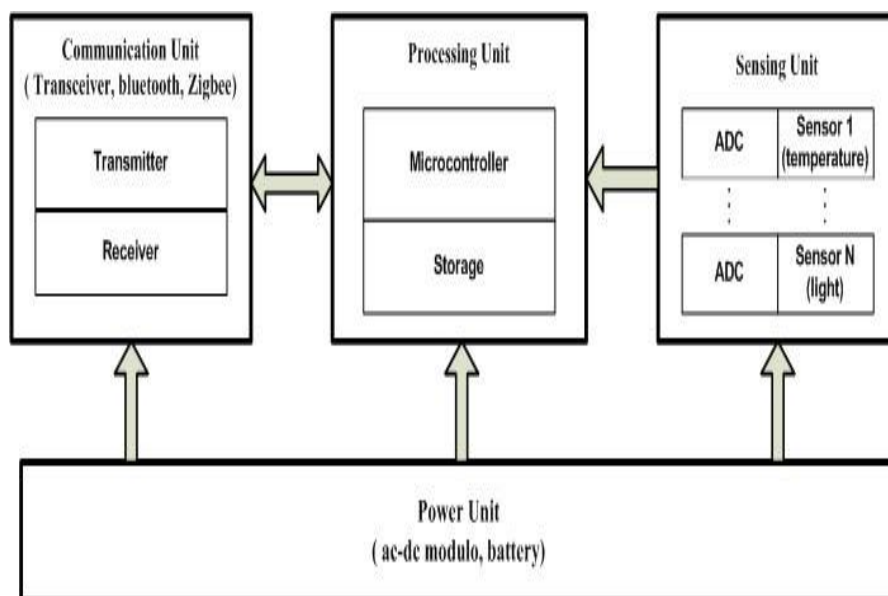


Figure 1.2 General hardware architecture of a sensor node (Akyildiz, Melodia and Chowdhury, 2007)

The major components of a sensor node have been described below:

- **Sensing Unit:** Sensing unit is the main component performing sensing operations of the node. The main difference between the sensing unit of a node and that of other embedded communication systems is that a sensing unit generally includes several sensors to enhance its ability to gather various types of data from the environment, e.g., temperature, pressure, etc. Apart from a sensing unit, each sensor includes an analogue-to-digital converter (ADC). While the sensing unit senses various environmental phenomena and produces analogue signals based on them, the analogue-to-digital converter converts these to digital signals for

further processing and communicating (Potdar, Sharif and Chang, 2009; Healy, Newe and Lewis, 2008).

- **Processing Unit:** The main component that controls the sensor node is the processing unit, consisting of a microcontroller and a small storage unit (Fengchao, 2011; Healy, Newe and Lewis, 2008). The microcontroller processes the data sensed by the node and also controls the functioning of its other components. It manages the procedures crucial for the node to be able to carry out sensing operations, run algorithms associated with these operations, and collaborate with other nodes in wireless mode. For example, Imote2 uses Intel's processor (Intel PXA271, with operating frequency of 400 MHz, 32MB program memory and RAM) mode (Potdar, Sharif and Chang 2009).
- **Communication Unit:** The communication unit provides a wireless interface to handle transmission and reception of data packets, e.g., transceiver, Bluetooth or Zigbee. The transceiver carries out the necessary procedures required to convert the bits into radio frequency, to enable them to be transmitted to, and then recovered, at the other end. It allows any two sensor nodes in the wireless network to communicate with each other. The radio in the transceiver of a sensor node can be used to operate it in four different modes: (i) idle, (ii) transmit, (iii) receive, and (iv) sleep (Gungor and Hancke, 2009).
- **Power unit** The power unit provides the sensor node with energy to work. Usually, it uses a battery but other energy sources, such as AC/DC and filtering, can also be used. The unit powers all the components of the sensor node and, in the view of its limited capability, its energy sources need to be energy efficient to enable all the components to perform their tasks (Fengchao, 2011; Xuejun, 2010; Healy, Newe and Lewis. 2008). For example, Atmega128 (Mica2Dot, Mica2) consumes 8mW (Mica2Dot consumes 1.08 nJ/instruction) and 33mW (Mica2 consumes 4.459 nJ/instruction) during active mode (Potdar, Sharif and Chang, 2009) .

While this section explained the sensor node hardware architecture, the next will elaborate the working of the sink node in a WSN.

1.2.3 Sink and Gateway Nodes

A WSN basically has three types of nodes: common, sink, and gateway nodes (Ruiz, Nogueira and Loureiro, 2003). While common nodes collect the sensed data and sink nodes receive, store and

process these data, gateway nodes connect the sink nodes to the external entities known as observers (Ruiz, Nogueira and Loureiro, 2003). The sink node may be called the head node as it gathers and controls the data collected by other nodes, as shown in Figure 1.1. Therefore, the sink node is more powerful with more capabilities than the common and the gateway nodes.

Having explained the sink node, the next section goes on to explain sensor node management in WSNs.

1.2.4 Sensor Networks Management

Sensor nodes are often deployed at remote and inaccessible geographical locations, making network maintenance or reconfiguration, or recovery from technical problems or failure, rather impractical. Therefore, a sensor node management system that can work continuously without needing too much human intervention, is of great importance for the successful working of a WSN (Ruiz, Nogueira and Loureiro, 2003). Figure 1.3 shows the network management architecture for a typical WSN.

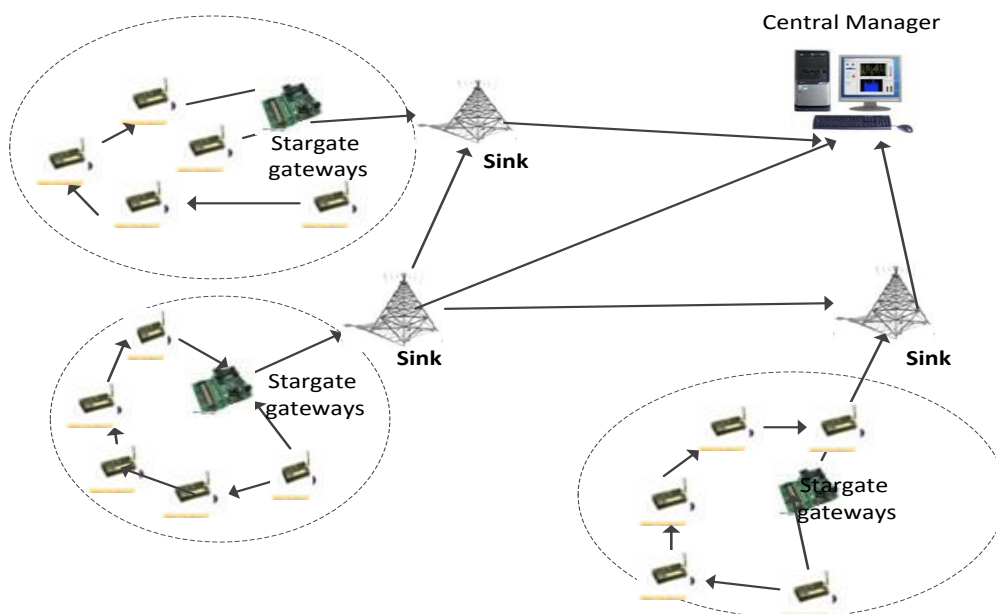


Figure 1.3 Network management architecture of WSN

On the basis of their network architecture, sensor network management systems are usually categorized into three types: centralized, distributed, and hierarchical.

-
- Centralized management systems have a base station acting as the managing station to collect data from all the nodes and control the entire network. However, this approach suffers from three disadvantages. One, the large volume of data pooling gives it a high message overhead, which limits its scalability. Two, the central server becomes the only point for data traffic concentration, increasing the possibility of failure of the network. Three, if the network is partitioned, the nodes that lose access to the central server, have no access to any management functionality either (Vivas, Fernández-Gago and Benjumea, 2010).
 - Distributed management systems have multiple managing stations. Apart from managing their own sub-networks, the managing stations also collaborate with each other to perform various managing functions in the network. However, this approach is complicated. Besides, the distributed management algorithms are often computationally too expensive to be used by the resource-constrained sensor network nodes (W Wang, 2008) (Wenjing and Younggoo, 2006).
 - Hierarchical management system may be said to be a combination of the centralized and the distributed approaches. It has intermediate managing stations to delegate management functions, with no direct communication among them. Each managing station manages the nodes of its own sub-network and passes on the data to the next higher-level managing station, while, delegating the management functions given to it by the higher-level managing station to its sub-network. Thus, this type of architecture brings in the benefits of centralized as well as distributed management approaches, and therefore, is the most suitable for a WSN (Vivas, Fernández-Gago and Benjumea, 2010).

After the background information about sensor network management systems, the next section discusses some of the real life applications of WSNs.

1.2.5 WSN Applications

A WSN is a network of many different types of sensors: e.g., magnetic, seismic, visual, thermal, radar, and acoustic sensors. As such, these sensors can monitor a broad range of environmental conditions: e.g. temperature and humidity, wind movement, speed and direction, light intensity, noise level, and certain kinds of objects present in the environment (Wen-bo, Hai-feng and Pei-

gen, 2010) (Potdar, Sharif and Chang, 2009). Therefore, WSNs can be employed in a wide variety of applications which can be broadly categorized into two types: Monitoring and Tracking. *Monitoring applications* can be used to monitor indoor as well as outdoor environmental conditions; health and wellness in the healthcare sector; power, inventory location and process automation in the industrial sector; and seismicity and structures in the construction sector (Yick, Mukherjee and Ghosal, 2008). *Tracking applications*, on the other hand, include the tracking of vehicles, human beings, animals, and many other different kinds of objects (Yick, Mukherjee and Ghosal, 2008). The following sections give a few of the many different applications of WSN that have been successfully tested in real, practical conditions:

- **Smart Dust:** One of the earliest projects in which the WSN found application was the Smart Dust (Pister, 2003; Warneke, 2001). The project aimed at providing sensing technologies for the military to be used in hostile environments. The use of the WSN extended the reach of these technologies, as it could be deployed in areas too dangerous to continuously operate in by the humans. A lot of critical information can be obtained for military uses by dropping a sturdy, self-configuring and self-organizing WSN in the war zone. Small Unmanned Aerial Vehicles (UAVs), commonly called drones, are the examples of such WSNs. As shown in Figure 1.4, a drone is an aircraft with no human pilots on board. The flight of a drone can be controlled in two ways: through computers on board the drone itself, or through a pilot based on the ground or flying in another vehicle, using remote control (Yick, Mukherjee and Ghosal, 2008).
-

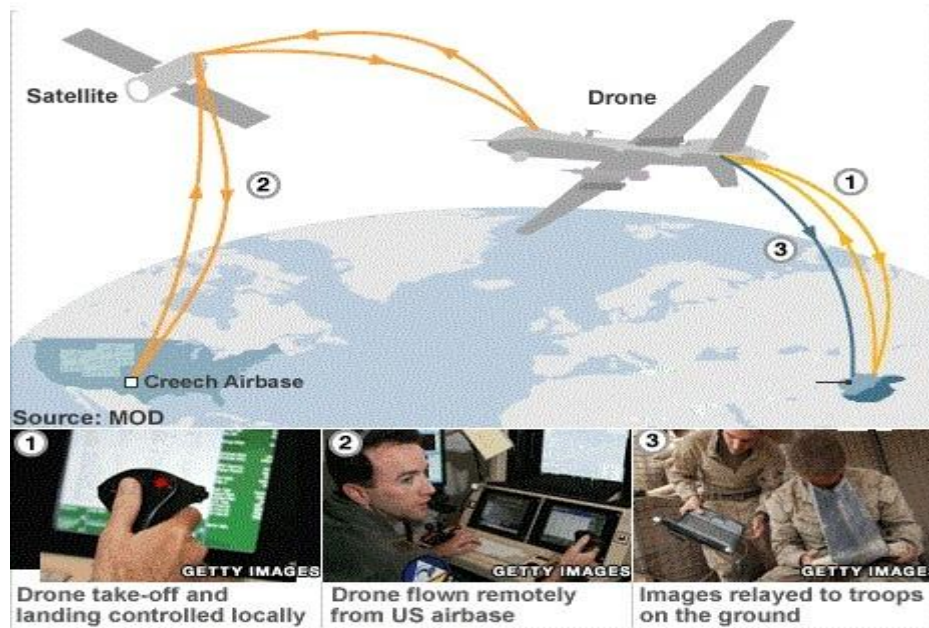


Figure 1.4 How drones work (BBC and Asia 2012)

Since drones are agile, fast moving, and can keep in constant motion, they have an enormous potential for use in the futuristic war systems (Abdul Hadi Fikri Bin Abdul Hamid, 2009). A drone has an elaborate imaging subsystem consisting of a number of enabling technologies, that include sensors and computing devices, apart from a wireless communication system. A typical drone has multiple digital cameras on board, with interfaces to a geospatial processor. A data networking system distributes the georeferenced imaging data, allowing for a simple and flexible system configuration. The control computer triggers the camera, stores images, prepares them for transmission, and also records data like camera setting, altitude and position. These data are attached to the images as metadata and then released to the ground station controlling the drone, through a state-of-the-art wireless network that can retrieve wireless data, including large files, real time (Ahmad and Samad, 2010). Drones can capture and stream multi-megapixel images in large formats, as well as the meta data. MQ-1B Predator and MQ-9 Reaper are two such drones currently being used in Afghanistan and Pakistan by the US (BBC and Asia 2012).

- Another example of the application of WSNs is the Redwood Microscope, a WSN case study to monitor and record the growth of redwood trees in the Sonoma area of California (Gilman, Joseph and Kevin, 2005). It uses a sensor node platform of a repackaged Crossbow Mica2 mote, with a form factor diameter of 1 inch. The mote had an Atmel ATmega128

microcontroller with a frequency of 4 MHz, a Chipcon radio with a frequency of 433 MHz and speed of 40Kbps, and a flash memory of 512KB, as illustrated in Figure 1.5 a. Each sensor node measures atmospheric temperature and humidity, as well as solar radiation that is photo synthetically-active. The sensor nodes are placed at different heights on the trees, as illustrated in Figure 1.5.b. The Redwood Microscope project aimed at helping plant biologists to track changes in the spatial gradients of the microclimate of a redwood tree, in order to validate the biological theories.

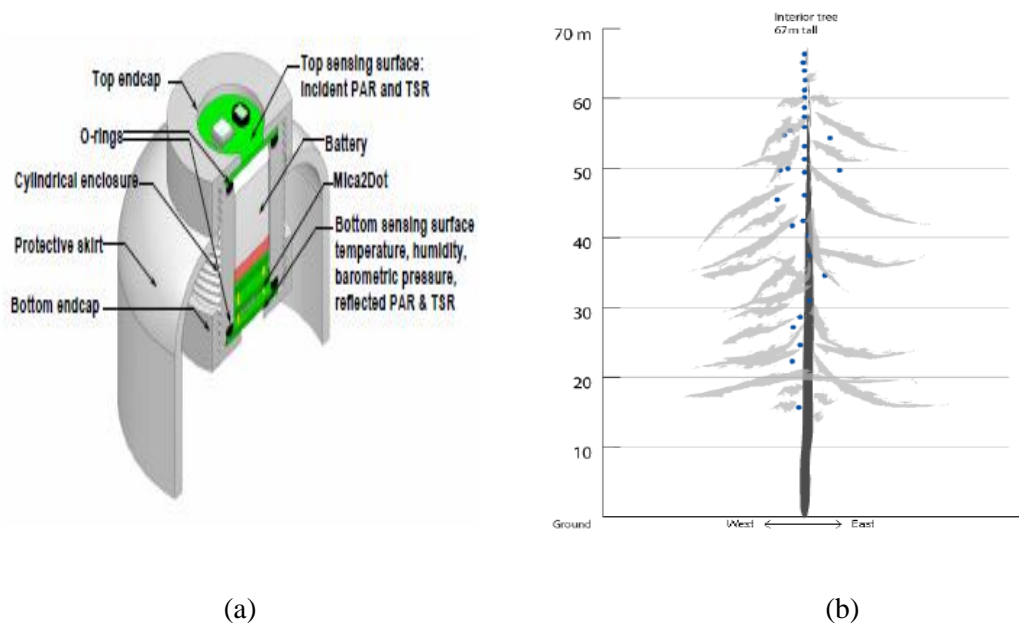


Figure 1.5 a. Repackaging sensor node b. Placement of nodes within a tree (Gilman, Joseph and Kevin, 2005)

- Akyildiz, Pompili and Melodia (2005) have investigated several key aspects of the underwater acoustic communication system. Further, Vasilescu, Kotay and D. Rus (2005) have developed an underwater platform for a sensor network system that can monitor fisheries and coral reefs in the long term. The system includes static as well as mobile sensor nodes placed underwater. The static system is underwater sensor node Aquafleck while the mobile system is autonomous underwater vehicle (AUV). The system is connected to an ultrasonic as well as an optical communication networking modality. Ultrasonic communication has long been used in underwater applications, especially in autonomous underwater vehicles. The AUV is used to dock and transport the Aquaflecks with a matching optical communication link, while the Aquaflecks themselves can be used

for data mulling. The data collection experiments were carried out at Starbug, in Moreton Bay, Brisbane.

- WSNs have been used for early flood detection and warning systems in several developing countries of Central America, particularly Honduras (Elizabeth, Sai and Daniela, 2008). The existing systems mainly aim at helping the personnel in continuous river bed monitoring. The system was developed at MIT and first deployed for tests in Honduras, where the urban life is significantly affected by frequent floods. For flood monitoring, sensors have to be deployed over a large area. The system network has a two-tier topology, as elaborated in Figure 1.6. There are three different sensors, to measure the atmospheric temperature, rainfall and water flow data respectively, in the lower tier of the network topology. The closely located sensors are grouped together, and connected to the second-tier of computation nodes. The computation nodes carry out data collection and processing. The results of the computation are then released to the third tier, i.e., the flood control centres. The system uses four different kinds of nodes which, along with their functions, have been described below:

- Sensing Node: This node measures the variables required to detect and predict the events of interest, such as atmospheric temperature, and rainfall and water flow data. This node collects data over a short frequency of time, e.g. minute to minute, and transmits them through a 900 MHz transceiver to the computation node.
 - Computation Node: This node basically serves two purposes. First, it feeds the dataset sent by the sensing node for prediction. Second, it communicates from the sensing nodes to the control centre via a 144 MHz transceiver.
 - Government Offices Interface Mode: It provides the users an interface to interact with the network through visualized data, and also enables network maintenance.
 - Community Interface Mode: This node receives the flood prediction data from the computation node or the government offices interface node, and informs the community under threat about the predictions.
-

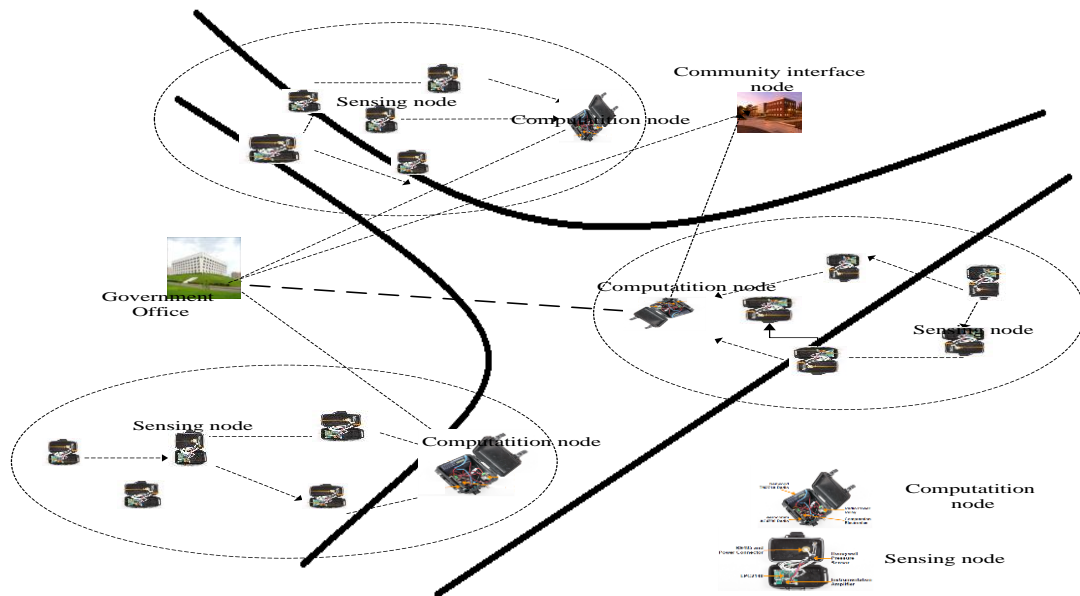


Figure 1.6 Sensor network for flood detection (Elizabeth, Sai, and Daniela 2008)

- WSNs have also found use in extreme terrains or conditions which are inaccessible to the humans. Volcanic monitoring (Werner-Allen, Lorincz and Ruiz, 2006) is one such example, carried out through a network of sensor nodes. WSNs can easily be deployed, installed and maintained for volcanic monitoring (Adrian, Szewczyk, J. D. Tygar Victor and Wen David, 2002) as equipment are small, light and power-efficient. However, a WSN application faces many challenges in volcanic data collection. These include reliability in event detection, efficiency in data collection, and ensuring high data rates in spite of sparse node deployment. To test the concept application, two cases studies were conducted in an Ecuadorian volcano during 2004-2005 (Werner-Allen, Lorincz and Ruiz, 2006), by running 16 nodes for a 19 day test. The nodes measured seismic and acoustic data, transmitting to each other at 2.4 GHz and back to the base station through a single repeater node at 900 MHz. The application aimed at collecting seismic information on the basis of the earthquakes occurring near the volcano.

WSN applications having been explained, the next section moves over to explaining Wireless Multimedia Sensor Networks (WMSNs).

1.3 Wireless Multimedia Sensor Networks - An Overview

Depending on their applications, WSNs can be classified into two types: Wireless Scalar Sensor Networks (WSSNs) and Wireless Multimedia Sensor Networks (WMSNs) (Manel Guerrero-Zapata, 2009). A WMSN is basically a network of audio and video sensors, commonly referred to as ASN, that are wirelessly interconnected and allow the retrieval of still images as well as audio and video streams, in addition to scalar data. The development of WMSNs has been greatly fostered because of their vast application opportunities based on their ubiquitous ability to capture multimedia content from diverse environments, as well as store, process real-time, correlate and fuse such content, received from heterogeneous sources. The easy availability of inexpensive microphones, image sensors and CMOS cameras has further accelerated this trend. The WMSNs have greatly enhanced the capabilities of the existing WSNs, and opened up new vistas of targeted applications, like environmental surveillance, industrial process control, and even traffic monitoring and rule enforcement, through the networks of multimedia surveillance sensors (Potdar, Sharif and Chang, 2011; Sharif, Potdar and Chang, 2009).

After this brief overview, the next section will explain the WMSN architecture.

1.3.1 WMSN architecture

In general, WMSNs have an architecture similar to that of WSNs, as described in Section 1.2.1, the only difference being in the kind of sensors. WMSNs consist of networks of heterogeneous multimedia sensors (video, audio, low image, and scalar sensors) with different data processing and transmission capabilities. Figure 1.7 shows a typical WMSN architecture. It senses multimedia data through multimedia sensor nodes and relays them through multimedia processing gateways that fuse and process the data. The multi-tier architecture approach uses heterogeneous sensors, as illustrated in Figure 1.7. The multi-tier architecture enables the sensors to process data at different levels, based on their memories, and computation and bandwidth capabilities (Akyildiz, Melodia, and Chowdhury 2007). The star gate gateway collects the multimedia data and transfers them to a sink. After the sink transfers these multimedia data, the users can manage them through satellite networks, or through the internet, and can even publish them or carry out data monitoring tasks.

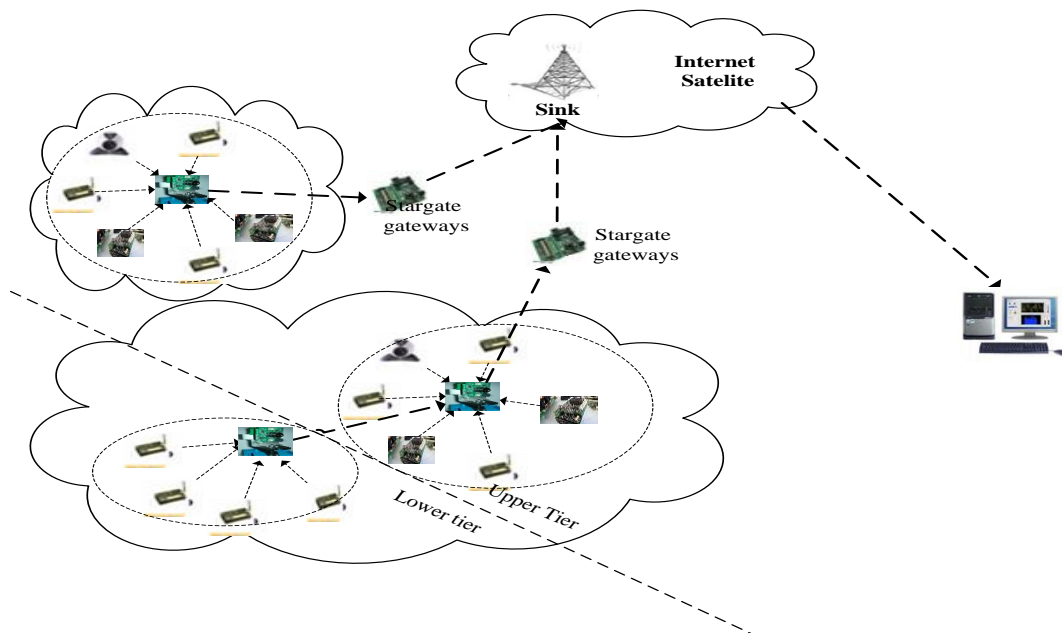


Figure 1.7 Wireless Multimedia Sensor Networks architecture

After this brief overview of WMSN architecture, the next section explains the Multimedia Sensor Node hardware architecture.

1.3.2 Multimedia Sensor Node Hardware Architecture

A multimedia sensor node device consists of a number of basic components, including sensing unit, central processing unit, communication subsystem, coordination subsystem, memory, and optional mobility/actuation unit, as depicted in Figure 1.8.

The major components of a multimedia sensor node have been described below in detail:

- **Sensing Unit:** As already explained in Section 1.2.2., a sensor node is capable of sensing scalar data. Going a step further, the sensing unit in a multimedia sensor node is capable of sensing not only scalar data but also video and audio streams. A sensor in a multimedia unit consists of two subunits: a sensor, which may be in the form of a camera or audio and/or scalar sensor, and an analogue-to digital converter (ADC). While the audio sensor captures sounds in the sensed event, the camera captures still or moving images. The typical resolutions for the camera are in terms of pixel/inch, and for the audio sensor in DB. The function of the ADC is to convert the analogue signals produced by the sensor, on the basis of the observed phenomenon, into digital signals.

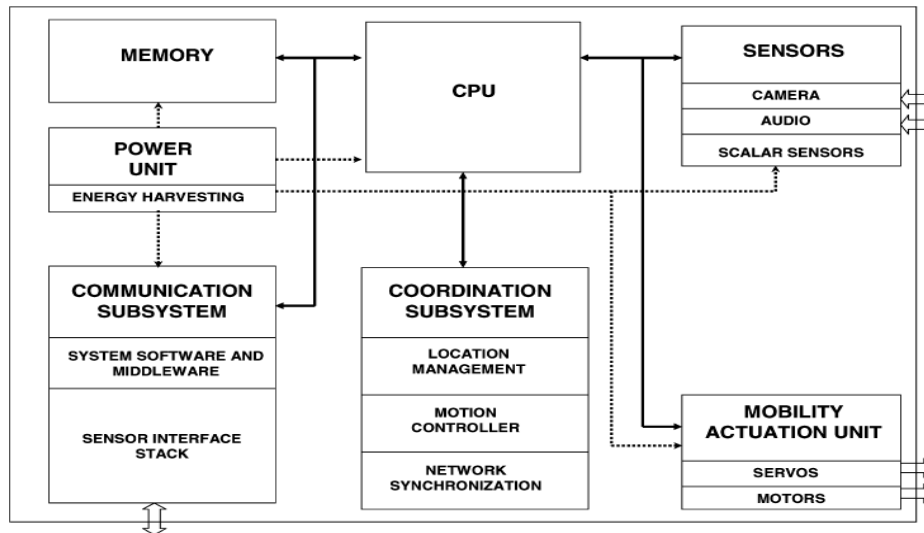


Figure 1.8 General Hardware architecture of multimedia sensor node (Akyildiz, Melodia, and Chowdhury 2008)

- **Central Processing Unit:** As mentioned in section 1.2.2, the Imote2 uses Intel's processor, Intel PXA271, which is the central processing unit (CPU) of Imote2, working as the principal controller of this wireless sensor node. However, the main controller of a multimedia sensor node also the system software. For example, the Stargate board, designed and produced by Intel and Crossbow respectively, uses Intel's PXA-255 XScale RISC processor operating at 400 MHz, with 32 Mbyte of flash memory and 64 Mbyte of SDRAM. This CPU, interfaced with a memory, coordinates the sensing and communication tasks by executing the system software in charge of this function. On the whole, compared to a scalar sensor node, a multimedia sensor node is substantially more powerful, and capable of sensing and retrieving audio and video streams.
- **Memory Unit:** The memory unit of a multimedia sensor node usually consists of both flash memory and RAM. For example, the Stargate board has 32 Mbyte of flash memory with 64 Mbyte of SDRAM (Sharif, Potdar and Chang, 2009). The flash memory contains the program code for the multimedia node while the RAM stores any information or data required for computation. Some of the memory units also have non-volatile storage for off-line data capture, to be retrieved later.
- **Power Unit:** As mentioned in Section 1.2.2, the power unit of a sensor node is the most important part of the hardware as it powers the whole system. It is supported by an energy generating unit, such as a battery of solar cells. The sensing arrays, such as the CCDs, or the

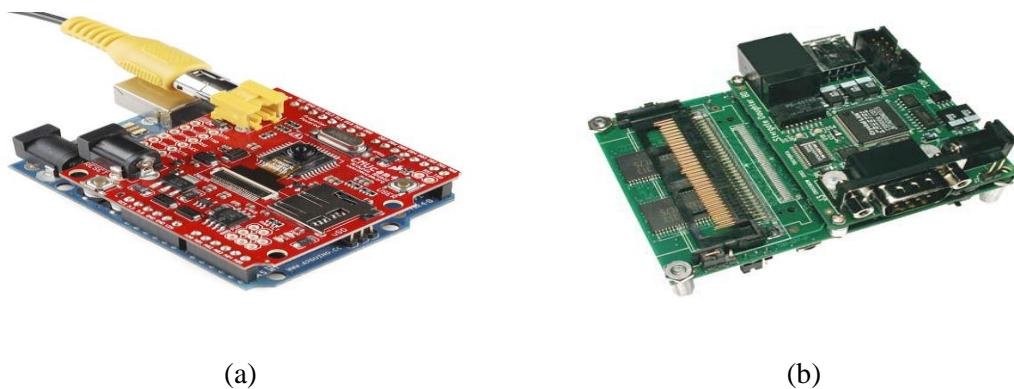
multimedia sensors (Akyildiz, 2007), such as the CMOS image sensors, generally require a lot of power. The amount of power consumed in the sensing subsystem of a multimedia sensor node is considerably higher than in an ordinary, scalar sensor. For example, a scalar temperature sensor consumes $6\mu\text{W}$ in sensing the environmental temperature. However, for image capturing, Cyclops consumes $42\mu\text{W}$, CMU-Cam consumes $42\mu\text{W}$ and High-end PTZ camera consumes 1W (Akyildiz, Melodia and Chowdhury, 2008).

- **Communication Unit:** The device is interfaced to the network through a communication subsystem which consists of a transceiver and communication software, including the communication protocol stack and the system software, for example an operating system and a middleware.
- **The Coordination unit:** The functioning of the different network devices is coordinated by a coordinating subsystem, carrying out operations like location management and motion control.
- **Mobility Actuation unit:** An optional unit in multimedia sensor node, this unit enables the node to move or manipulate the object, as per the requirements. A scalar sensor node has no actuator.

WMSNs use their various internal nodes to harvest not only scalar data, such as humidity, temperature, air pressure, light intensity, and various acoustic data, from the environment, but also multimedia information, such as audios/videos and digital images (Akyildiz, Melodia and Chowdhury, 2007). Thus, a WMSN has the imager as its main sensor, e.g. SparkFun CMUcam4, which can be seen in Figure 1.9 a. The main processor in SparkFun CMUcam4 is Parallax P8X32A (Propeller Chip) which is connected to an OmniVision 9665 CMOS camera sensor module, with a VGA resolution (640x480) RGB565/YUV655 color sensor, and an Onboard Image Processing system (QQVGA 160x120). The CMUcam open-source programmable embedded color vision sensors are low-cost, low-powered sensors, meant for mobile robots (Xiao, 2006). The Stargate 2, as shown in Figure 1.9 b, is a small form factor ($3.5'' \times 2.5''$), designed and produced by Intel and Crossbow respectively. It has Intel's PXA-255 Scale 400 MHz RISC processor with 32 Mbytes of flash memory, 64 Mbytes of SDRAM, and an on-board connector to connect it to

Crossbow's MICA2 mote, in addition to PCMCIA Bluetooth or IEEE 802.11 cards. It also has high processing ability along with more on board resources.

The visual data handled by a WMSN puts its sensor network under severe constraints. Collecting, processing and disseminating visual data is a process intensive activity and requires high bandwidth. However, WMSNs boast of many novel features as they have sensor nodes with video cameras, as well as high computation abilities.



This section described multimedia sensor node hardware architecture. The next section will shed light on some applications of WMSNs.

1.3.3 WMSNs Applications

Most of the existing and potential WMSN applications require the paradigms of the sensor networks to be rethought in order to arrive at a mechanism that can deliver multimedia content at the predetermined Quality of Service level. WMSNs have several existing applications and are expected to enable several other new applications, all of which can be classified into five categories (Akyildiz, Melodia and Chowdhury, 2008). A few examples of each deployed and tested in real life applications have been described below:

- Personal and Health Care: WMSNs can deliver, and enhance the quality of, resuscitative care by collecting and automatically integrating the patient's vital signs into patient care record, to be used real-time, as well as correlating them with the hospital records and the long term observations (Tao, Jingchun and Yonglei, 2001). A 3G multimedia network, incorporated into

a telemedicine sensor network (Fei and Sunil, 2003), is one such example that can provide health care services with a wide reach. Patients carry medical sensors that monitor different health parameters, such as blood pressure, breathing rate and body temperature. Going further, advanced remote monitoring can also be performed by medical centres using audio sensors, motion/activity sensors and cameras, all fitted in a wrist device worn by the patient (Fei and Sunil, 2003). Moreover, patients' behavioural practices can also be studied through multimedia sensor networks in order to pinpoint the cause of the illness. For example, the elderly people's behaviour can be studied to identify the triggers of conditions like dementia (Reeves, 2005).

- **Multimedia Surveillance:** WMSNs are, in fact, networks of interconnected video cameras that are battery-powered and miniature in size. Each audio/video sensor is connected to a low powered wireless transceiver that can sense, process and transmit video signals. This integration of sensor network and video technology forms the basis for a new generation, multimedia surveillance system (Sharif, Potdar and Chang, 2009), which allows the use of many different media (sensor signals, texts, images, audio and video) for a complete automatic analysis and interpretation of an environment on a real time basis, which can complement and enhance the existing surveillance systems for better protection against crime and terrorist attacks (Lin, Ming-Hua and Hsieh Tseng, 2009).
 - **Traffic Monitoring and Enforcement:** The transport system is one of the many sectors that are expected to benefit from the enhanced monitoring and surveillance enabled by WMSNs. For example, Kansas City began a pilot project on an intelligent transport system in September 2004 (NRI, 2012), which monitors around 75 miles of highways in the city. Such WMSN based systems can be immensely helpful in avoiding traffic jams, reducing emissions and saving fuel. Such systems can monitor vehicular traffic on highways or busy roads in big cities, and be integrated with services to offer alternative routing advice, avoiding congestion. Such monitoring of vehicular traffic can also be used to get useful information for the traffic system, such as number of cars and their average speed. Further, these systems can detect violations of traffic norms and transmit the relevant video streams to the law enforcement agencies.
-

-
- **Environmental and Industrial Applications:** WMSNs, equipped with acoustics sensors are ideal for identifying, investigating the causes of, and resolving circumstances, like unaccounted for variations in product quality in industries, or unusual noises or vibrations, or similar other signs of environmental problems. For example, oceanographers have used an array of video sensors and various image processing techniques to determine the process of evolution of sandbars (García Villalba, Javier and Sandoval Orozco, 2009). Multimedia content like imaging, apart from scalar data like temperature and pressure, can be used in time-critical process control in industries. For example, in the quality control process in manufacturing, the final products have to be inspected through an automated system to find defects. Another common industrial process that profits from WMSNs is the monitoring of machines for the purposes of diagnosis and preventative maintenance. For example, the pulp and paper industry employs complex mechanisms in its massive rolling machines, and even small variations in the alignment, temperature or speed of the roller can seriously affect the operation or the quality.
 - **Gaming:** Networked gaming has emerged as a popular entertainment activity today. WMSNs have been used in the prototypes of some futuristic games involving virtual reality, to enhance the gaming experience of the player, by incorporating sight and touch input, making the player respond to it as if live (Mauricio, Radenkovic Steve and Oppermann Adam, 2005). However, in gaming, there are strict constraints on multimedia data return, as the WMSN application heavily depends on the placement of the sensors, and it is not easy for the player to carry them without interfering with his gaming experience. However, the pace at which the popularity of these games is growing, there is little doubt that there will be further researches in designing and deploying pervasive WMSN systems that would make possible a richer interaction of the players with the game environment. *Can You See Me Now (CYSMN)* (Kundur, Unoma and William, 2006) is one such game utilizing WMSN, which integrates online and physical gaming.

Having described the applications of WMSNs in this section, the next section explains the proposed digital watermarking technique for secure communication between two sensor nodes, both in WSNs and WMSNs..

1.4 Digital Watermarking – An Overview

The rapid developments in communication technology, and especially the World Wide Web, have also increased the risk of piracy considerably. A situation has arisen when any kind of reproduction results in copies that are nothing more than degraded versions of the original object or the cover medium. The multimedia data are available on the Internet in digital form, which makes it very easy to reproduce the exact copies of the original. Moreover, the copying devices are quite low cost and efficient, and therefore, almost anyone can afford them. In addition to these factors, there are already many complex requirements for managing the illegal distribution of cover medium that financially damage the legal owner of the object.

Cryptography is not adequate to protect the cover medium which is publicly available but whose redistribution is unauthorized. However, digital watermarking is a technique that has the potential to solve this problem. Digital watermarking refers to the process whereby information such as hidden copyright notices or verification messages are added to the cover medium like digital audio/video, image signals, or documents, to protect the ownership rights (Potdar, Han and Chang, 2005; Wang, Xu and Yang; 2009). These hidden messages consist of a group of bits giving information about the signal or the author of the signal.

There are many different situations where the digital watermarking technique can be applied. The first of these that comes to the mind is the need to provide the proof of ownership (Jian and Xiangjian, 2005). When the cover medium is published on the web through open communication channels but the owners want to retain the copyright, they need to have a means of proving their ownership in case of a dispute. In general, a digital signature or watermark signal is embedded in the cover medium in a way that only the owner can extract it. If the pirates want to steal the cover medium property, they would have to extract the original watermark signal from the cover medium, and maybe, insert their own. As it is not easy to do so for them, the watermark comes in handy to verify that a certain copy of the cover medium is indeed copy, and that it has not been changed to an extent that “critically” alters its contents.

The digital watermarking techniques are designed to insert the watermark directly into the original signal or into some transformed version of the cover medium, in order to take the benefit of the perceptible properties or robustness of a particular signal manipulation. The digital watermarking

technique has two basic stages: stage one embeds a watermark signal using an embedding algorithm and an embedding key. Stage two uses a detection algorithm and an appropriate watermark detection key to retrieve the watermark signal. In most techniques, the embedding and detection keys are kept secret.

Figure 1.10 and Figure 1.11 show the watermark embedding and detection process respectively.

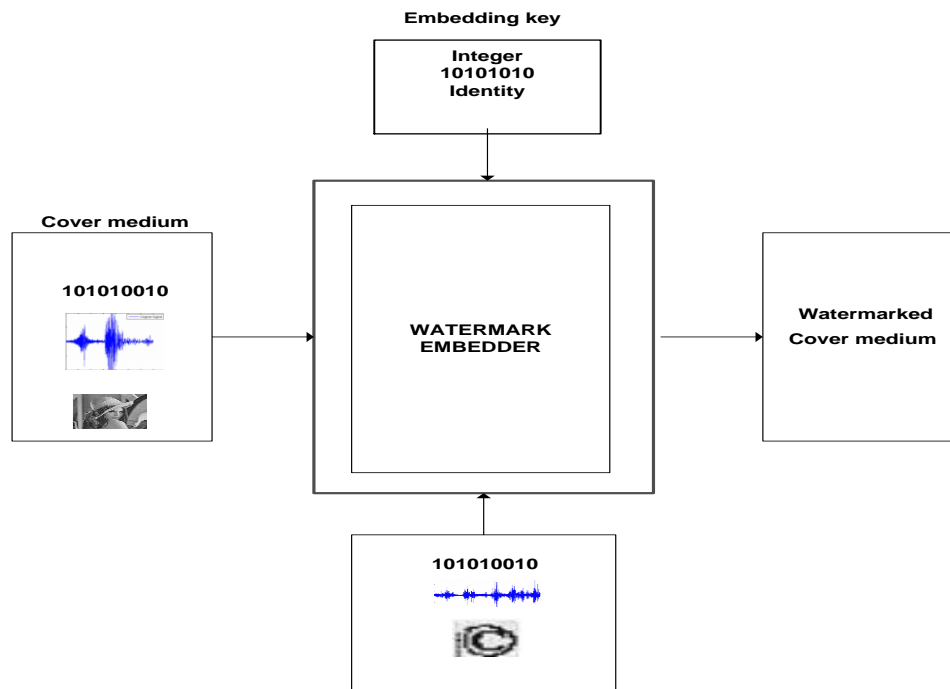


Figure 1.10 Watermark embedding process

As illustrated in Figure 1.10, the watermark signal is embedded, together with the embedding key, in the original cover medium. The watermarked cover medium is generated by a watermark embedder.

As further illustrated in Figure 1.11, the distorted watermarked cover medium, together with the original cover medium and the embedding key, is fed into the watermark detector, which recovers the watermarked cover medium and the original cover medium at the receiving end. The detection key is used to extract the watermark signal.

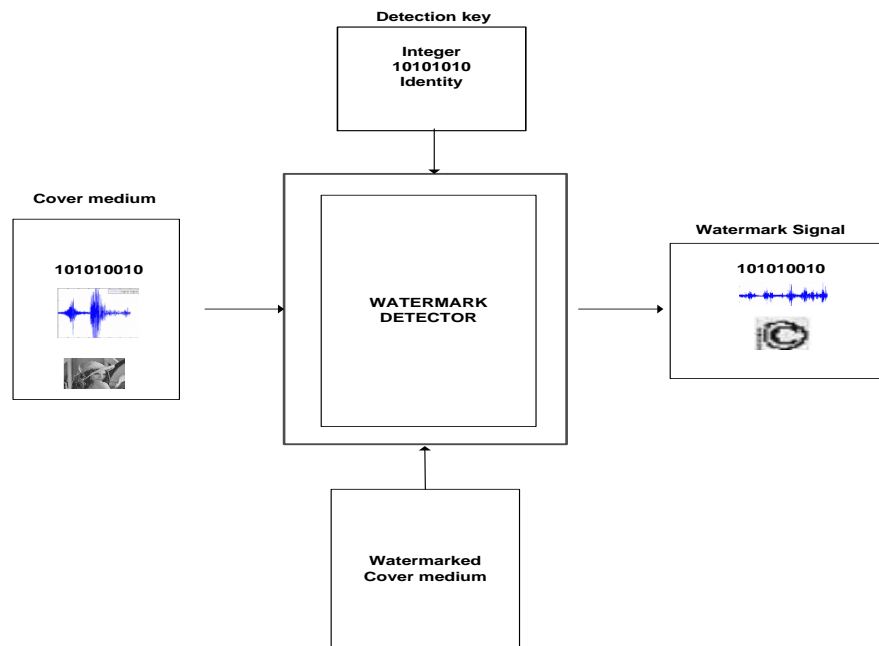


Figure 1.11 Watermark detection process

After this overview of the digital watermarking technique, the next section describes some applications of digital watermarking.

1.4.1 Digital Watermarking Applications

In general, the digital watermarking technique is divided into two types: robust and fragile. Robust watermarking can survive attacks, allowing the watermark signal to be detected when required. On the other hand, fragile watermarking breaks easily, even with slight modification in the cover medium. The subsequent sections describe the applications of both robust and fragile watermarking techniques:

Robust Watermarking

The main applications of robust watermarking include:

- **Broadcast Monitoring:** Broadcast monitoring is a technique to cross-verify, using watermarks, whether the content supposed to have been broadcast has really been, or not. The time and place of the broadcast are found out in this technique by detecting and recognizing the watermarks embedded in the object which is broadcast. It can thus help an artist or an organization find out the dishonest broadcast stations, or ensure that the work is not being broadcast by a pirate station (Potdar, Han and Chang, 2005).

- **Copyright Protection:** While producing a new work, the copyright information can be embedded as watermark. In cases of disputes regarding ownership, this watermarked copyright information can always be retrieved and provided as evidence (Jian and Xiangjian, 2005).
- **Digital Fingerprinting:** The digital fingerprinting technique can be used to identify the rightful owner of a digital content. Just like human beings who have unique fingerprints, the digital content owners too can create unique digital fingerprints on their content through watermarking. Even though a digital object has many different user fingerprints, it can only have one unique owner fingerprint in the form of the watermark for copyright (Potdar, Han and Chang, 2005). This watermark is generated by using some individual, user identifiable parameter as input.
- **Copy Control:** As the name implies, this application controls the number of copies that can be made from a cover medium. It is controlled by the recording equipment that reads the watermark and acts according to the watermarked instructions. For example, if the instruction is “copy one more”, the content will be copied and the new instruction will be “copy no more”. In case somebody tries to copy the content again, the command will be rejected by the recorder because the instruction now is “copy no more” (Jian and Xiangjian, 2005).

Fragile Watermarking

The main applications of fragile watermarking include:

- **Content and Integrity Authentication:** In many multimedia applications, there is a need to authenticate an object that might have been tampered with (Jian and Xiangjian, 2005). For example, in an image of a scene of crime, changing anything in the image, such as the car number plate, will mislead the investigation into suspecting someone other than the real criminal. Watermarking technique can be used to embed a particular message in the digital media, in order to indicate the author of the content, and authenticate its integrity (Noury, Mercier and Porcheron, 2000).
 - **Content Archiving:** The process of archiving digital content, e.g. still images, audios or
-

videos, can also benefit from watermarking through the insertion of a serial number or a digital object identifier into the content. The same process can also help classify and organize the digital content better. Normally, digital contents are identified by their file names which, however, is a very unreliable way of identification as a file name can be changed very easily. Embedding a serial number or an object identifier into the object itself minimizes the possibility of tampering with (Potdar, Han and Chang, 2005).

- **Tampering Detection:** Watermarking can help detect tampering with digital content through fragile watermarks. Destruction or degradation of the fragile watermark would indicate that the content has been tampered with, and cannot be trusted. Detection of tempering is of crucial importance in applications dealing with highly sensitive data, e.g. medical or satellite imagery. Similarly, fragile watermarking can detect tempering with digital images presented in courts of law as proofs, and thus can be used as a forensic tool (Potdar, Han and Chang, 2005).

After this overview of the applications of digital watermarking, the next section moves on to explain the digital watermarking process.

1.4.2 Digital Watermarking Process

As discussed previously, digital watermarking refers to the process whereby information, such as hidden copyright notices or verification messages, are added to the cover medium, like digital audio/video, image signals, or documents, to protect the ownership rights (Wang, Xu and Yang, 2009). These hidden messages are in the form of a group of bits giving information regarding the signal or its author. The signals are in the form of still images, audios or videos and, if they are copied, the message, or group of bits, is also copied along with them. The key processes of a generic digital watermarking system can be seen in Figure 1.12.

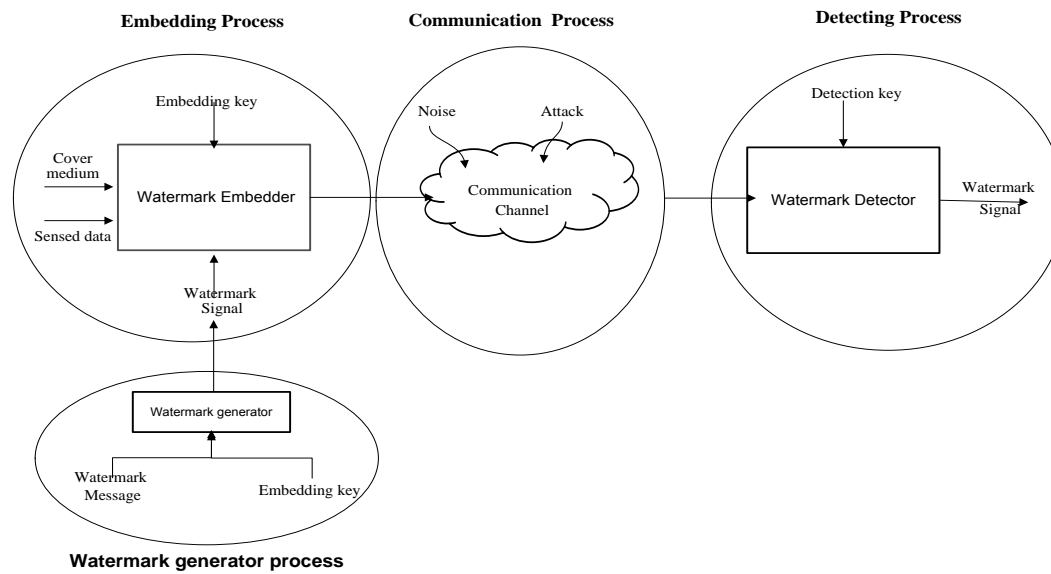


Figure 1.12 Key components digital watermarking system

As shown in Figure 1.12, a digital watermarking system as a communication task, involves four main processes, viz. watermark generation, embedding, communication and detection. The communication process includes blocking possible attacks, while the detection process includes watermark retrieval. The subsequent sections describe each of the processes shown in Figure 1.12.

1.4.2.1 Watermark Generation Process

Watermark generation process is the first and, a very critical, step in the watermarking process because of its unique and complex requirements. The information contained in the watermark message, whether text or sensed data, must be unique. The watermark key must also be unique. e.g. in the form of a binary stream, integer or amplitude, in order to keep it secret. Both the watermark message and the watermark key generator act as inputs, and are then processed in the watermark generator to produce a watermark signal. The hash function (Kamel and Juma, 2011) and the modulation pulse (Zhang, Liu and Das, 2008) can be cited as examples of watermark generators. The watermark signal is a kind of signal or pattern that can be embedded into the cover medium. There are two types of watermark signals, meaningful and meaningless watermark signals. Examples of meaningful watermarks are image logos, spread spectrum sequences, and permutations of watermarks. On the other hand, pseudo random sequences, M-sequences and chaotic sequences are meaningless watermark signals (Bai, Harms and Li, 2008). Now, we explain the watermark generation process, shown in Figure Figure 1.12 in greater detail:

1. **Watermark Message:** The information which the watermark signal carries is called the watermark message. It is usually communicated from the sender to the receiver. In some cases, there is a need to communicate only one bit (on-off signalling), but in others, usually an array of watermark messages has to be transmitted.
2. **Watermark Key:** A key is used to construct a watermark signal, which is then inserted into the cover medium. The watermark message, such as the signature of the owner, can be encoded using the standard cryptographic techniques, such as Rivest Shamir Adleman (RSA), Message Digest 5 (MD5) and Rivest Cipher 4 (RC4), in order to construct a watermark signal that is resilient against the statistical and pattern recognition attacks (Jessica and Potkonjak, 2003) (Koushanfar and Potkonjak, 2007).
3. **Watermark Generator:** It is the system that generates high quality watermark signals. The process uses both the watermark message and the key. A hash function, a modulation pulse or a median filter are examples of watermark generators (Padmavathi, Shanmugapriya and Kalaivani, 2010).

1.4.2.2 Embedding Process

The second stage of a watermarking system is the embedding process, undertaken by an embedder. The embedder combines the cover medium, the watermark signal, the sensed data, and the embedding key, and creates a watermarked cover medium. The watermarked cover medium, which is perceptibly identical to the cover medium, is then transmitted by the sender through the unsecure communication channels, such as wireless and radio channels. The subsequent sections will explain the watermark embedding process, as shown in Figure 1.12, in detail:

1. **Cover Medium:** The cover medium is the medium which a watermark signal is embedded in. Different kinds of data, images, audio/video signals and texts are examples of cover media. As explained earlier, the content providers want to insert watermarks into a cover medium for several reasons, such as copyright protection, broadcast monitoring, digital fingerprinting, copy control, content authentication, tampering detection, etc. (Potdar, Han and Chang, 2005). To take an example, commercial advertisements can be monitored through their watermarks to confirm their timing and count. Sometimes, images with a copyright registration symbol ©,
-

have their watermark removed by specialized software. In such cases, invisible watermarks can be used to overcome the problem.

2. **Embedding Key:** The embedding process requires a secret key to be used. This key must equal the detecting key, because it is this key which is known to both the sender and the receiver.
3. **Watermark Embedder:** Watermark embedding is the process of inserting a watermark signal into a cover medium. One of the characteristics of watermark embedding is imperceptibility which is one of its most important requirements, as the watermark has to be perceptibly transparent. Another name for this perceptible transparency is fidelity, referring to the extent of similarity between a cover medium with watermark and one without watermark. Although the watermarking system is based on the imperfection of the human eye (I. Cox, M. Miller and Bloom, 2002), in some watermarking systems the watermarks can be made visible to serve some purpose.

1.4.2.3 Communication Process

During transmission, if there is anything that interferes with the communication process, resulting in deterioration of the quality of transmission, such as noise, a watermarked cover medium may be lost. The other thing to reckon with are attacks, such as modification, tampering, and manipulation attacks. The aim of these attacks is to modify, delete or remove the watermark signal from the watermarked cover medium. More intricate details of the communication process, shown in Figure 1.12, are explained below:

1. **Communication Channel:** In the context of digital watermarking process, the process of embedding and extraction of the watermark is modeled as communication channel, whereby the watermark signal is distorted because of strong interference and channel effects. Strong interference is caused by the presence of the channel effects corresponding to the signal processing operations (Akan, Pascal Zhang and Qian Jayant, 2008).
 2. **Noise:** From the embedding to the extracting stages in the watermarking process, there can be many things interfering with the communication channel, all of which are referred to as noise. Noise can be external or internal, and can disrupt the communication process at any point by affecting the incoming data rate, causing the channel to drop packets and lowering the quality of data transmitted (Akan, Pascal Zhang and Qian Jayant, 2008).
-

3. **Watermark Attack:** Any action meant to harm a watermark signal is referred to as a watermark attack. Such attacks aim to remove or destroy all watermark signals from the cover medium, e.g. by modifying data, adding false data, replicating data, disturbing data sampling, tampering with data packets, and so on (Kamel and Juma, 2011).

1.4.2.4 Detection Process

The end of the watermarking system is the detection and extraction process, which is a very crucial process as the sender identifies and provides information to the intended receiver at this stage. The detection and extraction are undertaken by a detector. An extraction unit first extracts the watermark signal, and later compares it with the cover medium to detect the same. The extraction process consists of two phases: detection of the watermark information and extraction of the same. Depending on the requirement of the cover medium in the detection process, it can be of two types: if the original cover medium is required, it is called informed detection, and if not, it is called blind detection (Potdar, Han and Chang, 2005). A watermarking system with an informed detection process is also known as private watermarking, whereas a watermarking system with a blind detection process is also known as public watermarking. The watermark detection process, as shown in Figure 1.12, is described in detail below:

1. **Watermarked Signal:** The process of extracting the watermarked signal is undertaken in a watermark detector. The result of this process is a watermarked signal. This watermarked signal differs from the original watermarked signal since it has already gone through unsecure channels, which might have interfered with it by causing noise or possible watermark attacks.
 2. **Detection Key:** The detection process requires a secret key to be used. This key is usually kept equal to the key used in the embedding process, if secrecy is required (Jian and Xiangjian, 2005). A watermark system normally uses three types of keys: public key, private key, and detection key. While the public can extract the watermark signal using the public key, only the author can extract it using the private key, which can, therefore, be likened to the product author's signature.
 3. **Watermark Detector:** The process of extracting and detecting a watermarked signal from a cover medium is termed watermark detector. In some applications, the watermarked object
-

is compared with the unwatermarked object to authenticate the presence of a watermark, and to isolate the watermark signal. This type of watermark system uses informed detection (Potdar, Han and Chang, 2005). On the other hand, using a correlation measure to find out the strength of the extracted watermark signal requires blind detection (Potdar, Han and Chang, 2005).

After this description of the digital watermarking process, the next section explains the digital watermarking techniques for WSNs..

1.5 Digital Watermarking Techniques for WSNs

To explain how digital watermarking technique works in WSNs, we take a scenario of the integration of WSN and digital watermarking in a real life application. Basically, the aim will be to show how the scalar data captured by a WSN node is protected from various data attacks through watermark, embedded using the digital watermarking technique.

Recently, the State Electricity Company (SEC), the monopolistic electricity distribution company of the Government of Indonesia, suffered financial losses to the tune of 2.1 billion Indonesian rupiah (IDR) due to rampant electricity theft, both in the capital city of Jakarta (Detik News, 2012) and the rural areas, such as Surakarta (Harianjoglosemar News, 2012). Electricity theft is done by mounting electrical instruments on illegal electricity installations. The Mini Circuit Breaker (MCB), an illegal instrument used by the thieves, can hide the additional resources used by the customer. Therefore, the SEC needs to detect the use of this instrument and monitor energy consumption of the consumers, and also convince the consumers about the quantity of electricity supplied, in order to prevent electricity theft. By deploying WSNs, i.e. wireless networks of autonomous sensor devices, distributed over a space to collaboratively monitor physical phenomena, the SEC can monitor the amount of electrical energy used by its consumers, and substantially reduce the financial losses.

For example, consider a scenario in which WSN nodes are deployed in a residential area suspected of stealing electricity, to automatically report the cases of theft to the SEC, as illustrated in Figure 1.13.

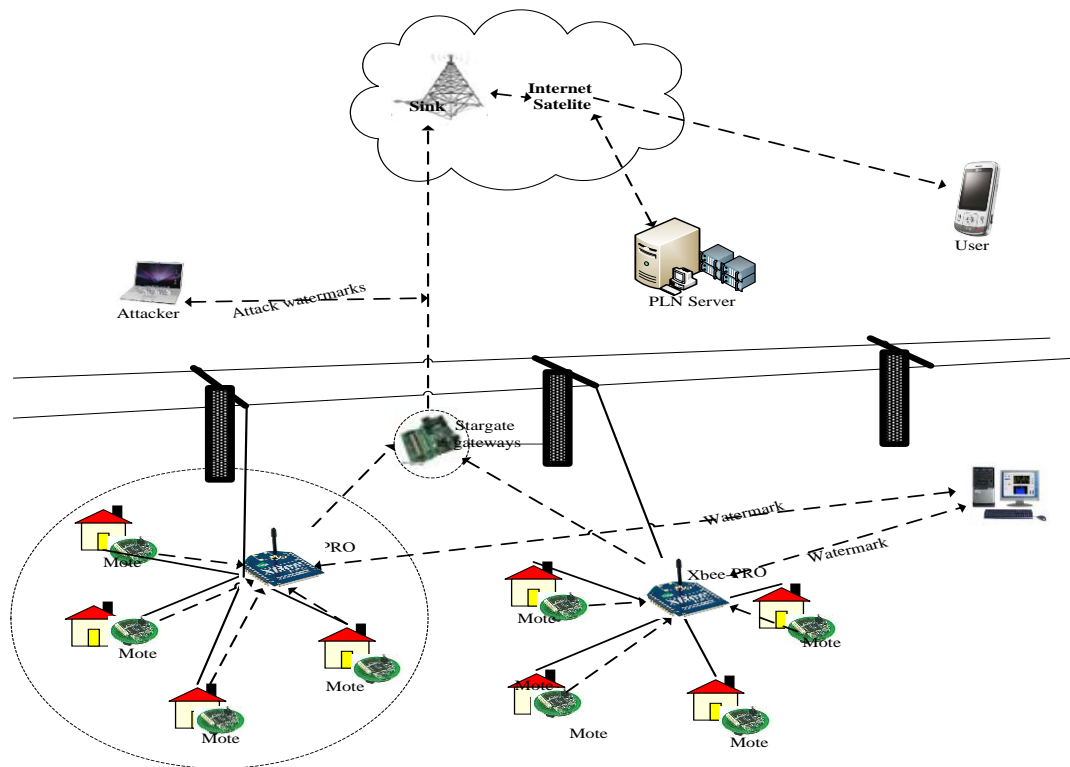


Figure 1.13 An example of the use of watermark in WSNs

Each sensor node senses the object and then sends the data object wirelessly to the Xbee-PRO. The objects are voltage, current frequency and phase difference between some points, which are sensed in the distribution channel, with each point in the channel giving electrical energy to some house. Voltages and currents are sensed at the ADC input of the internal WSN node. Data from the ADC, i.e. the frequency and the phase difference, along with time of storage, is stored on an SD Card. After collection, the data are sent wirelessly to the gateway using the Xbee-PRO. Furthermore the data sent to the sink through the stargate gateway are stored on a SEC server. The data are published by the SEC on the Internet to let the consumers know the quality of the electrical energy supplied, via mobile phone. To prevent the theft of electricity by the consumers suspected of stealing, the SEC hires a commercial entity that provides original data from them.

Let us assume that the WSN node and the Xbee-PRO are managed by a commercial entity with a commercial intent, i.e. to provide the original data to the SEC. This commercial intent induces the entity to maintain the original data in a way to prove its authenticity as and when the need arises. Hence, it embeds watermarks into the data.

Suppose an attacker manipulates the watermarked data during the transmission process and sends the manipulated data to the SEC through the Internet. The commercial entity can still prove that the data is tampered with using the watermark. The scenario can be depicted in Figure 1.13.

With this explanation of the digital watermarking technique for WSNs, the next section moves on to explaining the digital watermarking technique for WMSNs.

1.6 Digital Watermarking Technique for WMSNs

Similarly, to explain how the digital watermarking technique works for WMSNs, we take a scenario of the integration of WMSN and digital watermarking in a real life application. Again, the aim will be to show how the image captured by a WMSN node is protected from various kinds of attacks through watermark, embedded using the digital watermarking technique.

Lately, crime statistics pertaining to mini-supermarket thefts have increased in Indonesia, both in the capital city of Jakarta and the rural areas, such as Surakarta. A supermarket works through the day, and remains open till late night, without adequate security. In addition, most of the supermarkets in Indonesia have no Closed Circuit Television (CCTV) cameras making it all the more easy for the thieves. The Indonesian police have recommended using CCTV cameras for surveillance and monitoring by the supermarkets. If they are installed, these surveillance and remote monitoring systems can be deployed using WMSNs at critical locations, apart from businesses, such as airports, railway stations, military compounds and airbases.

Further, let us consider a scenario in which WMSN nodes are deployed in the Central Business Area on Slamet Riyadi Road (as shown in

Figure 1.14) to monitor and control crime by automatically reporting such events to the governing authority. Let us also assume that the WMSN is managed by a commercial entity with commercial interests, so that when a criminal activity is recorded by the WMSN nodes, the commercial entity has a strong incentive to ensure that it owns the footage, to prove its case in the court of law. Hence, it embeds watermarks to prove the ownership of the content. The advantage of watermarking is that every recorded footage has an invisible/visible footprint of ownership which is difficult to remove. Additionally, the commercial entity can also add fragile watermarks to protect against tampering. If attacks are targeted during the transmission, the fragile watermarking

will be destroyed, indicating that the video footage is tampered with. The scenario has been illustrated in Figure 1.14.

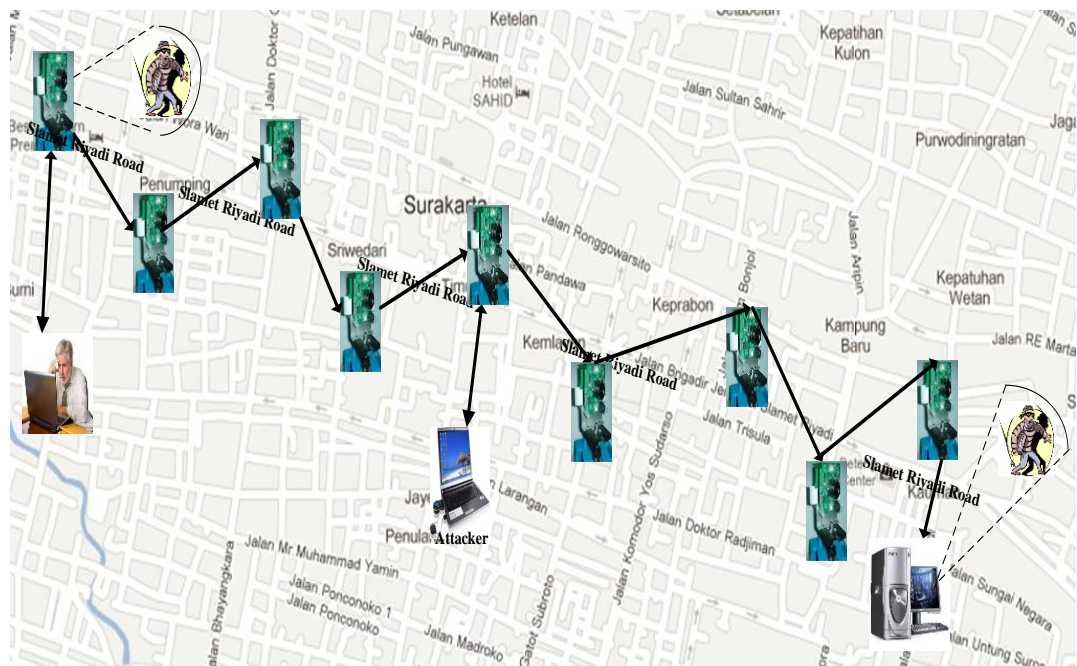


Figure 1.14 An Example of the use of watermark in WMSNs

The development of WMSNs has received a boost because of the availability of inexpensive CMOS image sensors and microphones these days, along with the wide range of their application opportunities based on their ubiquitous ability to capture multimedia data from different environments. In coming years, WMSNs are expected to not only enhance application opportunities of the existing sensors, such as environmental monitoring or tracking (Leigh, Renambot and Johnson, 2006), but also lead to the development of many new applications, e.g., in telemedicine and healthcare of the elderly or disabled, by pinpointing the causes of their illnesses, such as dementia (Reevesm, 2005). Other such applications can be recognition and localization of services provided to the users, and process control in manufacturing industry.

WMSNs undoubtedly possess many novel features, mainly because their sensor nodes are equipped with video cameras and have high computation abilities. However, they have also brought new security challenges in their wake. Security is an important issue with WMSNs because they are prone to several types of intentional network attacks, like man-in-the-middle attack (Kamel and Juma, 2011), and also because they suffer from the effects of bad network channels (Honggang Wang, Dongming Peng and Wei Wang, 2008). As a result, the authenticity of

the data transmitted cannot be verified. The man-in-the-middle attack can modify the transmitted data in various ways, e.g. by altering or deleting it, or by inserting extraneous data to it, while a bad network channel may introduce noises into the signal causing data damage. Addressing the challenge of these attacks is important to ensure a secure and trustworthy WMSN network. However, a WMSN node has very limited power supply in the view of its high computational ability, and hence, using a strong cryptographic algorithm is difficult. Therefore, watermarking techniques have become the most sought after ways to address these challenges , like preventing tampering and proving ownership.

Hence, research in the area of watermarking in WMSNs is becoming increasingly important. With the emergence of the concept of cyber-physical systems, i.e., on the web of things, this research has come to the main stream and has assumed a manifold significance.

1.7 Research Motivations

The main motivation for this study of digital watermarking technique for both WSNs and WMSNs comes from the fact that it can address the challenge of a secure communication between the sensor nodes of both WSNs and WMSNs. Digital Watermarking techniques have been shown to be useful in addressing issue copyright protection. The main stimulus for studying digital watermarking techniques for both WSNs and WMSN networks, therefore, is provided by its potential in addressing the issues,

1. Need for copyright protection of scalar data in WSNs.
2. Need for copyright protection of images in WMSNs.

1.7.1 Copyright Protection of Scalar data

The application of WSNs in sensitive areas, such as military surveillance, health and wellness in the healthcare sector, inventory location and process automation in the industrial sector, and seismicity and structures in the construction sector (Yick, Mukherjee and Ghosal, 2008), has a fundamental requirement of a secure scalar data transmission mechanism. As seen in in sub section 1.5, the SEC is becoming increasingly concerned about the rampant electricity theft in certain areas. To automatically report the cases of theft to the SEC, WSN nodes are required to be

deployed over the area to collaboratively monitor the physical phenomena. However, unlike the wired networks, malicious intruders have a high possibility of accessing the valuable sensor scalar data in wireless networks. The pirates can easily and limitlessly copy the digital data. Therefore, the copyright protection problem of the wireless sensor networks is a matter of concern while sending data in wireless transmission environments. Digital watermarking is a favourable technique to protect digital information and has widely been used in many secure applications. However, the copyright protection problems of WSNs are not sufficiently studied. The available solutions may not protect the copyright of the sensed scalar data. For instance, a malicious intruder may duplicate segments of the valuable sensor data for profit. Therefore, copyright protection becomes urgent and necessary.

1.7.2 Copyright Protection of Images

In sensitive areas, such as traffic monitoring and enforcement, or personal and health care, there is a fundamental requirement of a WMSN application to make the image transmission mechanism secure, as malicious attackers can easily access the sensitive multimedia content from the wireless network and make limitless copies of it. The digital watermarking technique has been widely deployed to secure many such applications and protect their digital information. For example, Pingping, Yao Jiangtao and Zhang Ye (2009) have devised a watermark algorithm for wireless sensor networks to provide real time copyright protection to JPEG images. Their method utilizes a CMOS image sensing system to embed the watermark in DCT domain.

However, at present, the existing solutions for copyright protection of images in WMSNs through digital watermarking are still fragmented, making more research an urgent requirement.

1.8 Research Objectives

This research primarily aims at proposing a conceptual model for copyright protection of scalar and multimedia data communicated through sensor networks, using digital watermarking. This will require an evaluation of the related literature, the development of solutions to specific research issues, the construction of models for these solutions, and the validation of one of the models using real-world data. This main aim of the research gives rise to two objectives as follows:

Objective 1: Developing a digital copyright protection system for scalar data in WSNs, using digital watermarking technique.

Objective 2: Developing a digital copyright protection system for multimedia data in WMSNs, using digital watermarking technique.

1.8.1 Developing Digital Copyright Protection of Scalar Data in WSNs using Watermarking Technique

This is the first objective of the research and specifically aims at developing a solution to address the issue of copyright protection of scalar data in WSNs. There are several open issues with copyright data protection in WSNs. The current secure copyright data protection models use Pretty Good Privacy (PGP) that contains public key encryption, hash function, and MD5, to provide a signature. The PGP requires a vast amount of computing resources in order to perform its operations. However, as sensors are resource constrained, the PGP algorithm is not well-suited for WSNs. A watermark adds a second line of defence to ensure that the data is valid, even if someone cracks the encryption. This thesis is going to develop a watermarking technique for copyright protection of data, based on the Linear Feedback Shift Register (LFSR) and Kolmogorov Rule (KR). LFSR is one of the methods of forming binary sequences for generating watermarks, and KR attempts to determine the length of the shortest binary computer program for the sequences.

1.8.2 Developing Digital Copyright Protection of Multimedia Data in WMSNs using Watermarking Technique

This is the second objective of the research and specifically aims at developing a solution to address the issue of copyright multimedia data protection in WMSNs. Again, there are several open issues with copyright multimedia data protection in WMSNs. The current secure copyright protection system for Joint Photographic Experts Group (JPEG) images uses the Discrete Cosines Transform (DCT) model, which is a feature of DCT coefficients produced through experiments. Watermark is embedded into low-frequency coefficients of the DCT. However, there is no system of copyright protection of images in WMSNs. Therefore, this thesis develops a watermarking technique for copyright protection of images, based on the Gaussian Pyramids (GP) and Kolmogorov rule (KR). GP is a method used in image processing that creates a stack of

successively smaller images when used multiple times. KR also attempts to determine the length of the shortest description of the sequences.

1.9 Significance of the Research

The significance of the issues addressed in this research is three-fold and the benefits resulting from it encompass the social, economic and technical (scientific) aspects.

1.9.1 Social Significance

1.9.1.1 Improvement in Theft Prevention

WSN nodes can be used in residential areas where theft of electricity is suspected to prevent such thefts, by automatically reporting the cases to the electricity distribution company. Placing WSN nodes can not only prevent theft of electricity but also convince the consumers about the quality of electricity. With LKR watermarking technique which is resulted from this theses, for WSNs being used by the commercial entity involved, the SEC need not fear theft of electricity.

1.9.1.2 Improvement in Societal Safeguarding

WMSN nodes can be deployed at critical locations, such as airports and railway stations. Placing WMSN nodes for surveillance and remote monitoring can help check crime and terrorist attacks to a great extent. With GPKR watermarking technique, which is resulted from this theses, for WMSNs being used by the commercial entity involved, the people at such critical locations will also feel safe as constant surveillance and monitoring will significantly reduce criminal activities

1.9.1.3 Improvement in Copyright Protection

Applying GPKR watermarking technique, which is resulted from this theses, for WMSNs can ensure ownership safety between nodes. By inserting watermarks, like invisible or visible footprint, the ownership to every multimedia record can be protected, with a commercial entity undertaking the WMSN operation with a commercial intent in proving the ownership. Thus, the attackers who modify and remove watermarks will be detected, and there will be no doubt as to the ownership of an image captured by a WMSN node during the the image transmission between the multimedia nodes

1.9.2 Economic Significance

1.9.2.1 Reducing Financial Losses

The SEC has already suffered financial losses to the tune of 2.1 billion rupiah, due to electricity theft. In order to reduce such financial losses, the WSN can be equipped and deployed over the area where theft is suspected, with the aim of preventing electricity theft and convincing the consumers about the quality of electricity. The LKR watermarking technique for WSNs will be operated by a commercial entity which will monitor electricity used by the consumers and give authentic data to the SEC, so that it can reduce its financial losses due to electricity theft

1.9.2.2 Creating Financial Advantage

WMSN nodes can be deployed in Central Business Areas and operated by a commercially entity. These nodes can capture the images of criminals or perpetrators of terrorist activities. The images can then be inserted with a watermark by using GPKR watermarking technique, in order to prove the ownership of the content. The commercial entity can then either report or sell it to the governing authorities. If attacks are targeted during the transmission, the fragile watermarking will be destroyed, indicating that the video footage is tampered with.

1.9.3 Scientific Significance

Many existing studies on digital watermarking technique in both WSNs and WMSNs have focused on the availability, confidentiality, authentication, and integrity of the copyright protection of data. This research aims at facilitating secure multimedia data communication and copyright data protection between sensor nodes in WMSNs, emphasizing the basic design principles that are expected to survive the currently going on rapid changes in WMSN technology.

- a. To the best our knowledge, this research is the only one of its type using the Linear Feedback Shift Register (LFSR), Kolmogorov Rule and Non Linear System Programming (NLSP) for copyright protection of scalar data in WSNs. The LFSR creates a binary stream by using a particularly key, which is then synchronized by applying the Kolmogorov Rule, resulting in the watermark constraints embedded in the Non Linear System Programming (NLSP). The
-

TOMLAB is used to solve the NLSP that creates the watermarking solutions for copyright protection in WSNs.

- b. Again, to the best our knowledge, this research is the only one of its type using the Gaussian Pyramids (GP) Transform for the image sensory and Kolmogorov Rule (KR) for copyright protection of multimedia data (images) in WMSNs. The GPT is used to reduce the image, to get the reduced image, and then also to expand the reduced image. The reduced image is synchronized by applying the Kolmogorov Rule, resulting in a watermark constraint. These watermark constraints are embedded in the NLSP. The TOMLAB is used to solve the NLSP that creates the watermarking solutions for copyright protection in WMSNs.

1.10 Structure of the Dissertation

The rest of the dissertation follows the following structure:

Chapter 2: provides the mathematical background to the preliminary digital watermarking techniques, together with security requirements and types of attacks, for both WSNs and WMSNs, and presents a comprehensive survey of the recent related research on digital watermarking for WSNs and WMSNs.

Chapter 3: defines the two research problems, based on the insights gained from the literature review.

Chapter 4: gives an overview of the solution and the conceptual process.

Chapter 5: presents the LKR watermarking technique for copyright protection of the scalar sensed data in WSNs.

Chapter 6: presents the GPKR watermarking technique for copyright protection of images in WMSNs.

Chapter 7 : brings the thesis to a conclusion by summarising its achievements and major benefits, identifying the work that still needs to be done, and laying down the lines for the future work in this research field.

1.11 Conclusion

This thesis covers areas in advanced ICT (Information Communication and Technology) which are not very well-known, and therefore, there are many concepts, terms and phrases that have to be introduced. Therefore, this chapter provided a brief introduction to the concepts related to the applications of digital watermarking technique for both WSNs and WMSNs, and the three technical fields namely WSNs, WMSNs, and the digital watermarking technique. It illustrated how these digital watermarking techniques for both WSNs and WMSNs can be used for copyright protection, content ownership management, secure and confidential communication, and tempering detection. It also highlighted the differences between the digital watermarking technique and the traditional security techniques, such as cryptography. Finally, it listed the main objectives and advantages of the proposed research and ended with the outline of the thesis. The next chapter provides a mathematical background to the preliminary digital watermarking techniques, together with the security requirements and types of attacks for both WSNs and WMSNs. It also surveys the literature relevant to digital watermarking technique for both WSNs and WMSNs, and gathers all the knowledge required to form a solid conceptual foundation and help the readers understand the technology proposed in the thesis.

CHAPTER TWO

LITERATURE REVIEW

This chapter covers

- ▶ preliminary concepts,
- ▶ security requirements for WSNs and WMSNs,
- ▶ attack models for WSNs and WMSNs,
- ▶ security requirements for digital watermarking,
- ▶ attack model for digital watermarking,
- ▶ copyright protection of WSN and WMSN data,
- ▶ survey and evaluation of digital watermarking techniques in WSNs, and
- ▶ survey and and evaluation of digital watermarking techniques in WMSNs

2.1 Introduction

This chapter explains the work done previously to address the issues outlined in Chapter 1. It also explains some theoretics in order to understand these works. These theoretics are atomic trilateration, Linear Feedback Shift Register, Kolmogorov Rule, and Gaussian Pyramid. The security requirements of WSNs and WMSNs, including their attack models, as also the preliminary digital watermarking techniques, together with their security requirements and attack models, have also been presented with the same end in view. The importance of the chapter lies in the fact that it gives an overview of the literature relevant to the field, and evaluates currently the most advanced digital watermarking techniques for both WSNs and WMSNs. Substantial progress has been made in the direction of providing a practical basis for the solution of a number of problems associated with copyright protection and content authentication. Nevertheless, there are several issues that need further investigation.

2.2 Theoretical Background

This section provides the theoretical background to this study of the copyright protection of scalar and multimedia data using digital watermarking to help facilitate a better understanding of the thesis.

2.2.1 Atomic Trilateration

Atomic trilateration can be used to generate a Non Linear System Programming (NLSP), also called a cover medium. With atomic trilateration in a two-dimensional sensor network, a multimedia sensor node in the network can be used to determine its position, by using the positions of, and distances to, at least three other multimedia sensor nodes of a known location. From these distances and positions, a multimedia sensor node trying to determine its location can generate a non-linear system equation. A typical scenario of atomic trilateration has been shown in Figure 2.1. Here, the sensor node D trilaterates with three other sensor nodes A , B and C . The coordinates of these nodes are (x_A, y_A) , (x_B, y_B) and (x_C, y_C) .

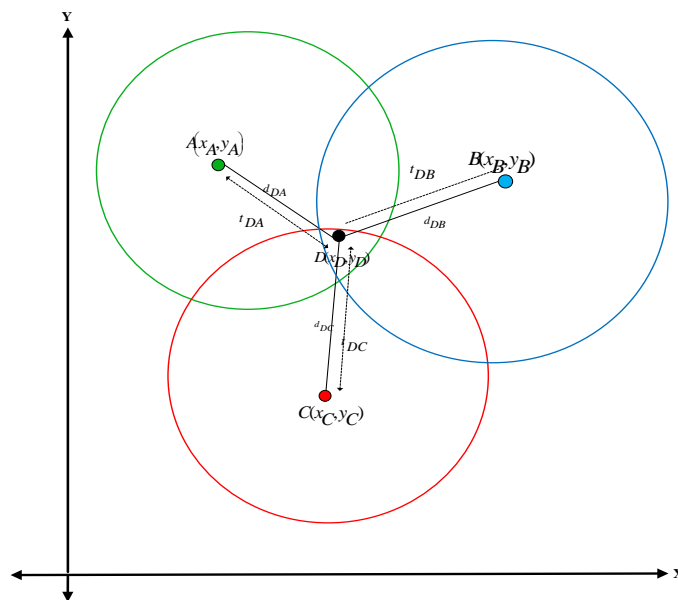


Figure 2.1 Atomic trilateration process

The distance is computed using the time differences of arrival (TDoA) between the acoustic signals emitted simultaneously from a multimedia sensor node and received at the node D , and the radio frequency (RF). The multimedia sensor node D turns on a timer upon receiving the RF signal from

a sensor node to measure the difference between the arrival of the RF and the acoustic signals from that multimedia sensor node. However, the timer measurements have an error. The speed of the acoustic signal is a function of the temperature of the propagation media. The relationship between the speed of the acoustic signal V_s (m/s) and the temperature T_c is as follows:

$$V_s = 331.4 + 0.6T_c \quad (2.1)$$

Assuming that the exact timer measurements between the node D and the sensor nodes are t_{DA} , t_{DB} , and t_{DC} , the distances of the sensor nodes with the node D , i.e. d_{DA} , d_{DB} and d_{DC} can be measured using TDoA as follows:

$$\begin{aligned} d_{DA} &= V_s * t_{DA} \\ d_{DB} &= V_s * t_{DB} \\ d_{DC} &= V_s * t_{DC} \end{aligned} \quad (2.2)$$

Then the coordinates of $D(x_D, y_D)$ are calculated by using Euclidean theorem

$$\begin{aligned} d_{DA} &= \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} \\ d_{DB} &= \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} \\ d_{DC} &= \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} \end{aligned} \quad (2.3)$$

Computing V_s from equation (2.1) and then replacing the values of the distances from equation (2.2) in equations (2.3), yields the coordinates (x_D, y_D) . However, there are errors in measuring T_c in equation (2.1), and in measuring t_{DA}, t_{DB} and t_{DC} in equation 2.2. These inaccuracies change the nature of the problem from a non-linear system of equations with a unique solution, to a problem of optimization where the objective is to reduce the errors in the system of equations to the minimum.

The next step is the formulation of equations (2.1), (2.2) and (2.3) as a non-linear optimization problem, in terms of $(\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3)$. The terms $\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$ denote the errors in the measurement of T_c, t_{DA}, t_{DB} , and t_{DC} respectively. Also, the variables $\delta_1, \delta_2, \delta_3$ are the errors in the Euclidean distances measured in Equation (2.3) and the TDoA distances d_{DA}, d_{DB}, d_{DC} , measured in equation (2.2) respectively. The objective of the function is to minimize the overall error in the system, and can be stated as shown in equation (2.4). This system is called an NLSP.

Objective function:

$$\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3$$

Constraints

$$\begin{aligned} \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) &\leq \delta_1 \\ \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) &\leq \delta_2 \\ \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) &\leq \delta_3 \end{aligned} \quad (2.4)$$

To make it clearer, atomic trilateration is the process used by the Global Positioning System (GPS) to map the co-ordinates. We know that the formula of distance between two GPS nodes requires speed and time. GPS signals travel with the velocity of light. In order to keep the system informed of the time when a signal is relayed from the satellite, every satellite in the GPS system has an atomic clock keeping accurate time. The GPS receivers too keep record of the time to monitor when the signal reaches the receiver, although their records are not as accurate as those of the satellites using atomic clocks. The distance is, therefore, obtained by multiplying the velocity of light, and the difference between the time of the satellite relaying the signal and that of the receiver

receiving it. Each satellite relays not only its identity and position but also the time of transmission. Besides, it also relays information on the location of other satellites. Comparisons between satellite signals in GPS systems are made by carrying out calculations shown above. The principle of trilateration (the determination of a distance from three points) is then used to calculate the distances between the satellite and the receiver.

This section explained atomic trilateration which is used to generate NLSP. The next section explains the Linear Feedback Shift Register (LFSR).

2.2.2 Linear Feedback Shift Register (LFSR)

Applying a linear feedback shift register (LFSR), whose characteristic polynomial is primitive, is one method of forming binary sequences for generating watermarks (Jian and Xiangjian, 2005; Harjito, 2003). LFSR can be seen as a shift register with its input bit being its previous state's linear function. Single bits can only have the exclusive-or (*xor*), as their linear function. Therefore, LFSR is a shift register with its input bit being driven by the *xor* function of some bits from the overall value of the shift register:

$$s_{K+n} = \sum_{i=0}^{n-1} c_i s_{k+1} \quad (2.5)$$

$$s_{K+n} = \sum_{i=0}^{n-1} c_i s_{k+1}, \text{ where } k \geq 0, n \in \mathbb{Z} \text{ and the } c_i \text{ are binary constants such that } c_0 = 1.$$

Associated with such a recurrence relation is a binary polynomial

$$f(x) = c_0 + c_1x + \dots + c_{k-1}x^{k-1} + x^k \quad (2.6)$$

called the characteristic polynomial of the LFSR. The coefficients c_i are feedback constants. Such sequences can be mechanized by using a linear feedback shift register (LFSR) whose tap settings are defined by the feedback constants.

The sequence u_i defined by the recurrence relation $u_{t+6} = u_t + u_{t+1} + u_{t+4} + u_{t+5}$ has characteristic polynomial $f(x) = 1 + x + x^4 + x^5 + x^6$ for this sequence, with

$$\begin{aligned} u_6 &= u_0 + u_1 + u_4 + u_5 \\ u_7 &= u_1 + u_2 + u_5 + u_6 \\ u_8 &= u_2 + u_3 + u_6 + u_7 \\ &\dots \end{aligned} \quad (2.7)$$

It can easily be shown that the first string is simply a sequence of 33 couples of 01. The second string looks like a random one but is not; it is the beginning of the binary expansion of the decimal part of 01. Therefore, the second string also has a simple description.

Kolmogorov complexity rule is used for numbering the variables of a linear combination in the optimization objective function, and the set of constraints (Jessica and Potkonjak, 2003), (Koushanfar and Potkonjak, 2007).

We know that the Kolmogorov complexity rule is the short description length of the overall description interpreted by the computer. Two papers (Jessica and Potkonjak, 2003), (Koushanfar and Potkonjak, 2007) have used the Kolmogorov complexity rule constraints for numbering the variables of a linear combination in the optimization objective function and a set of constraints.

After this introduction of Kolmogorov complexity rule, the next section introduces the Gaussian Pyramids.

2.2.4 The Gaussian Pyramids

This section explains the process of reducing and expanding an image, using the Gaussian image pyramid technique (Adelson, Anderson and Burt, 1984; Choudhary, Sinha and Shanker, 2012). This technique can be used for detecting a target pattern. It can also be used to design a data structure that supports an efficiently scaled convolution. This, it does by reducing the dimensions of the image, leading to a series of copies of the original image. The sample density as well as resolution of the image are decreased in this series of copies in regular steps. Now the process of reducing an image and expanding the reduced image will be explained

The Process of Reducing Image

The first step in Gaussian pyramid generation begins with an original image g_o which is then low-pass filtered to get a 'reduced' image g_1 . The operation can be continued to obtain a set of images g_o, g_1, \dots, g_n forming a pyramidal image structure, with each 'reduced' image having less resolution and sample density than the corresponding previous image. The low-pass filtering is done using a procedure that can be equated to the process of convolution by a set of local and symmetric weighing functions. The image sequence g_o, g_1, \dots, g_n is referred to as a Gaussian pyramid.

A fast algorithm outlining the process of generation of a Gaussian pyramid is given below and illustrated in Figure 2.3:

- The array g_0 represents the image consisting of R rows and C columns of pixels, denoting the zero level in the Gaussian pyramid.
- Level 1 of the pyramid consists of the image g_1 , which is a reduced, or low-pass filtered, version of g_0 . At this level, each value is a weighted average of the values at level 0.
- The level 2 is represented by g_2 . The values at this level are obtained from the values at level 1, by following the same weightage pattern.

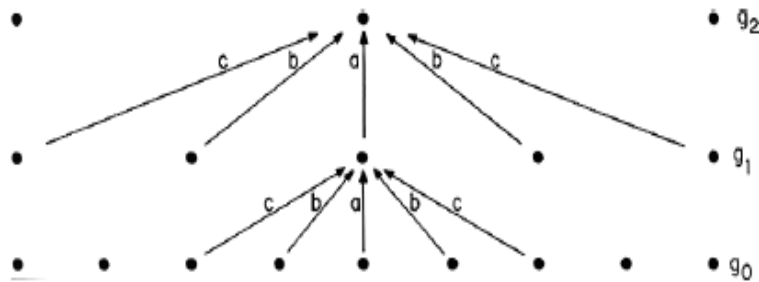


Figure 2.3 A graphical representation of the process in one dimension

The function `PYR_REDUCE` performs the process of level-to-level averaging, as follows:

$g_k = \text{PYR_REDUCE}(g_{k-1})$ which means that, for level $0 < l < n$, $0 < l < m$ and nodes i, j , $0 \leq i \leq C_i$, $0 \leq j \leq R_p$ the pyramid generation process can be represented as:

$$g_i(i, j) = \sum_{n=-2}^2 \sum_{m=-2}^2 w(m, n) g_{i-1}(2i + m, 2j + n) \quad (2.8)$$

where n is the number of levels in the pyramid, while R_p and C_j represent the dimensions.

Figure 2.4 shows the process of reducing an image.

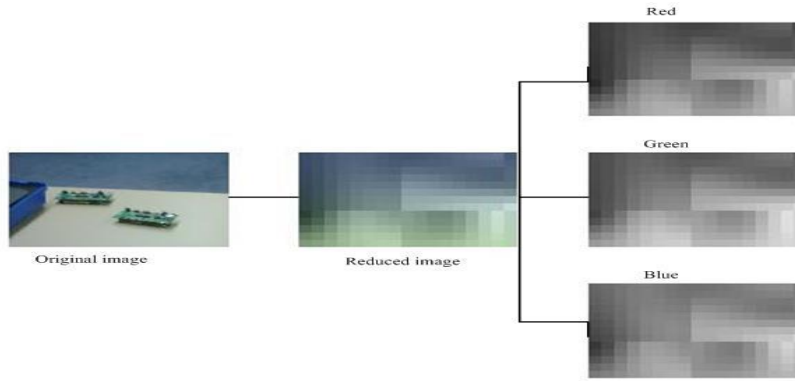


Figure 2.4 An example of reducing an image

The Process of Expanding the Reduced Image

Along with the Gaussian process of pyramid generation, there is also the reverse process of pyramid generation that expands a reduced image to its original scale. For the purpose, the function `PYR_EXPAND` is used, which is the reverse of the function `PYR_REDUCE`. It expands an $(M + 1)$ -by- $(N + 1)$ array into a $(2M + 1)$ -by- $(2N + 1)$ array. This is achieved through the interpolation of the new node values. Therefore, when the function `PYR_EXPAND` is applied to the array g_l of a Gaussian pyramid, it forms the array $g_{l,1}$ whose size is the same as that of the array g_{l-1} .

Suppose that expanding g_i by n times results in g_{l-n} . Now, $g_{l,0} = g_l$ and $g_{l,n} = \text{PYR_EXPAND}(g_{l,n-1})$. The function `PYR_REDUCE` applies to the levels $0 < l < N$, $0 \leq N$ and nodes i, j , $0 \leq i \leq C_i$, $0 \leq j \leq R_p$

$$g_{l,n}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l,n-1} \left(\frac{i-m}{2}, \frac{j-n}{2} \right) \quad (2.9)$$

The function `PYR_EXPAND`, applied once to the image g_l yields $g_{l,1}$ whose size is the same as that of the original image g_o (Adelson, Anderson and Burt, 1984).

The Figure 2.5 shows the process of expanding the reduced image to the original image

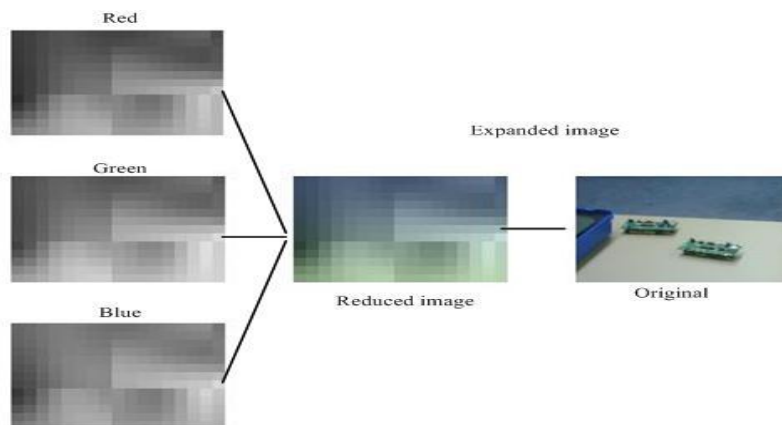


Figure 2.5 An example of expanding the reduced image to the size of the original

After this introduction of the preliminary concepts required to understand the related literature, the next section introduces the security issues in WSNs and WMSNs.

2.3 Security in WSNs and WMSNs

As pointed out in Chapter 1, WSNs face severe resource constraints. This is because their nodes are very small in size and have limited energy, resulting in low storage space, and limited processing ability as well as communication bandwidth. On the other hand, the video cameras and high computation abilities of the nodes give rise to new security challenges in WMSNs. Therefore, the security system, whether in WSNs or in WMSNs, must be able to protect the resources as well as the information communicated through the network, from both attacks and the misbehaviour of the nodes themselves. Therefore, this section discusses the security requirements for WSNs and WMSNs.

2.3.1 Security Requirements in WSNs and WMSNs

In order to be able to devise an appropriate security mechanism to ensure the protection of data and safety of the system, it is essential to understand the requirements of security in WMSNs in greater depth. In case of WSNs, the more critical are the constraints imposed by the limitations of memory, energy resources, and computational capacity. On the other hand, in case of WMSNs, complex algorithms need to be combined with the privacy and security policies so that the

multimedia content can be processed, compressed and distributed. If WMSN-based services are to be widely deployed in real life applications, it is imperative to reformulate the whole WMSN paradigm by incorporating a privacy and security policy as a fundamental requirement, which every sensor application must strictly adhere to, so that an adequate security level can be achieved. The following sections delineate these security requirements for WMSNs:

2.3.1.1 Confidentiality

In the context of WSNs and WMSNs, confidentiality means protecting the content communicated between the sink(s) and the sensors, e.g. commands, reports, or other multimedia data gathered by the sensors. Confidentiality here implies that any adversary with the privilege of access to the content must not be able to decipher or decode the messages exchanged in the network.

Being confidential here means protecting sensitive information contained in the data, so that no unauthorized third party can read it. The chances of a third party accessing and reading such data arise while the data travelling between two sensor nodes of a network, or between the sensor nodes and the receiving station. Confidentiality is essential in both WSNs and WMSNs, as an adversary with suitable resources can access and read sensitive data during these stages, in both types of networks, including many different types of sensed data or routing information. Such stolen data can be used for illegal purposes causing severe damage. Therefore, confidentiality conceals the data from a passive or an active attacker, so that any information communicated through a WSN or a WMSN remains protected.

2.3.1.2 Availability

Various threats to both WSNs and WMSNs may damage parts of a network, disable some of its functions or discontinue the services it provides, making it unavailable to the authorized users. The requirement of availability means ensuring that the essential functions of a WSN or WMSN, or the services it provides, remain unaffected even when they are attacked.

The requirement of availability makes sure that a multimedia node can use the resources, and that the network can be used to communicate data or information, making it possible to use the services or access the data when required. Sensor networks are vulnerable to many types of risks impairing availability, e.g. capture of a sensor node or denial of service attack. An impaired availability can make many critical operations and real time applications redundant. Hence, it is of crucial

importance to make WSNs and WMSNs resilient to any attack targeting their availability and keep alternatives ready in case of capture or disabling a node, for example by transferring its functions to another pre-determined node in the network.

2.3.1.3 Integrity

Integrity in the context of WSNs and WMSNs means that, if the data sent by the sensor nodes or the content sent by one user to another, has undergone any alteration, the receivers can detect it. It also means ensuring that such content or data are not lost, or they are not counterfeit, stale, or replicated old data.

Although the requirement of availability ensures that the data can always be accessed when desired, it cannot ensure that these data are complete and authentic. They might well have been changed, or some part of them might have been deleted, during transmission over the network. It is the requirement of integrity which ensures that these things do not happen. Integrity is a very important requirement as the use of wrong data or information may lead to disastrous results; in healthcare sector, for example, endangering lives. Therefore, implementing integrity controls is of critical importance to a WSN or WMSN. In short, it is required to rule out any tampering with, or alteration of, data or information, and ensure that it is reliable.

2.3.1.4 Authentication

The requirement of authentication in a WSN or WMSN means ensuring that the communication over the network is genuine. Authentication controls in such a network verify whether the data or information being communicated has emanated from a legitimate user. This is important because a malicious entity may have injected counterfeit data or resent the same data over the network. This may happen, for example, in case of a compromised node.

The X.800 (Castelluccia, Mykletun, and Tsudik 2005) guidelines recommend two layers of authentication, authenticating the peer entity and authenticating data origins. Authenticating peer entity means authenticating all the nodes participating in the communication. It can apply to two nodes that communicate with each other, or one node (e.g., the cluster head) that communicates with many other nodes around it (i.e, broadcast authentication) (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002). Broadcast authentication can be done at the sink or at an intermediary node where data is aggregated.

In attacks on communication networks, eavesdroppers can also inject data. Therefore, it is imperative for the receiver to ensure that the data going into decision-making processes has originated from the right sources, and hence the importance of authenticating data origins. Data authentication mechanism should be able to prevent unauthorized users to use the networks by detecting their messages and rejecting them. Appropriate data authentication controls are of critical importance for a WSN or WMSN to behave correctly (Grieco, Boggia, and Sicari 2009). To sum up, data authentication ensures reliability of data or information by identifying its origin, and verifying the sender and the receiver. It can be ensured by using symmetrical or asymmetrical mechanisms, whereby the sending and the receiving node have shared secret keys. However, ensuring data authentication is an extremely challenging task because of the distributed nature of WSNs and WMSNs.

2.3.1.5 Non repudiation

Non-repudiation in WSNs and WMSNs means ensuring that a node receiving a particular message cannot deny having received it, nor can the node sending it deny having sent it. The X.800 (Castelluccia, Mykletun, and Tsudik 2005) recommendations term the former as destination non-repudiation and the latter as origin non-repudiation.

In the context of information security, non-repudiation mechanism ensures that a transferred message has actually been sent by the party that claims to have sent it, and received by the party that claims to have received it, and both of them do not deny their actions of sending and receiving later on.

2.3.1.6 Data Freshness

Data freshness mechanism in a WSN or WMSN checks how old the data captured and sent by the nodes are. Data freshness is important as, even if data integrity and confidentiality are assured, the data need to be recent to be useful. Data freshness mechanism verifies it, and ensures that the data received are not old data resent by an attacker.

Data freshness may be strong or weak. A strong freshness mechanism provides total message order on a request-response pair with an estimation of delay. It is used for time synchronization within a network. On the other hand, weak freshness mechanisms provide only partial order,

without estimation of delay, to be used by sensor nodes (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002) .

2.3.1.7 Privacy

The requirement of privacy recognizes that the data being transmitted are sensitive and need to be sufficiently safeguarded. Such data usually come from telemedicine or military surveillance systems.

Both WSNs and WMSNs deal with a large amount of varied data. Some of these may directly or indirectly reveal some personal or otherwise sensitive information pertaining to the users. If malicious entities gain access to these data, they may use them in ways that violate or offend individual privacy. In the view of its sensitive nature, privacy is a prerequisite for many WSN and WMSN applications. No wonder it has attracted a lot more research than other requirements. Examples of works on privacy are ‘A Security and Privacy Survey for WSNs in e-Health Applications’ (de los Angeles Cosio Leon 2009), ‘Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications’ (Al Ameen, Liu, and Kwak 2012), and ‘Security and Privacy Issues with Health Care Information Technology’ (Meingast, Roosta, and Sastry 2006). A number of privacy solutions and policies have been suggested, such as secure data cloaking (Kundur, Unoma, and William 2006), secure communication channel (Douglas, Fidaleo, and Nguyen 2004), (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002), and definition of privacy policies (Karlof, Sastry, and Wagner 2004). However, they all cater to some specific aspects of privacy and none of them provides a complete privacy solution for WSNs and WMSNs.

This section explained the aspects of security in both WSNs and WMSNs that need to be taken care of in case of any attack. The next section will discuss different types of attack possible in both WSNs and WMSNs.

2.3.2 Types of Attacks in WSNs and WMSNs

WSNs and WMSNs are susceptible to many different kinds of attack which can take place in many ways. The most notable is the Denial of Service (DoS) attack, but attacks may also take the form of physical capture or destruction, violation of privacy, and so forth. DoS attacks can also take many different forms, ranging from jamming the network’s communication channels to more advanced attacks against any other aspect of security. The general category attacks are classified as shown in Figure 2.6.

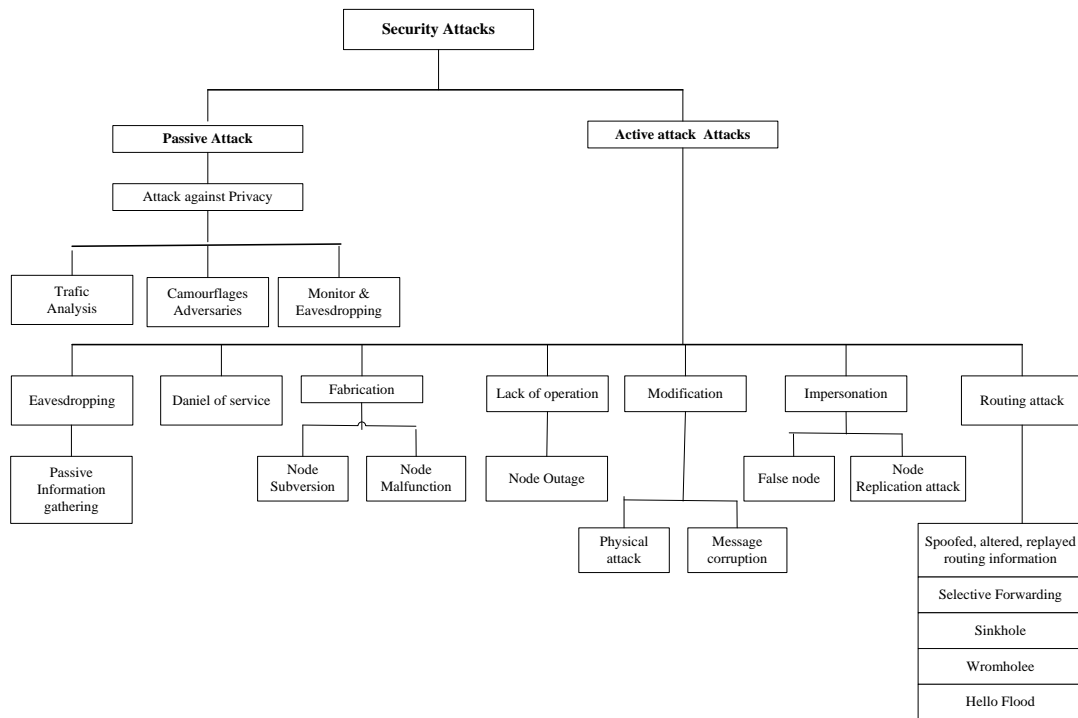


Figure 2.6 General classification of security attacks (Lingxuan and Evans 2003)

Potential asymmetry in power and computational constraints often makes it almost impossible to guard a network against DoS attack, if the attack is well planned and sophisticatedly carried out. A sensor node can be easily jammed by a more powerful node, effectively preventing the network from carrying out its functions. Moreover, DoS attacks are not the only types of attack on WSNs, they include many other types of attack, e.g. attack on sensor nodes, or routing protocol, or even on the physical security of a node. Many studies have been done on various types of attacks on sensor networks and the corresponding security mechanisms. For example:

- Han et al. (Han, Chang, and Gao 2006) present various attacks on WSNs in a systematic way.
- Roosta et al. (Roosta, Shieh, and Sastry 2006) present a comprehensive taxonomy of security attacks on WSNs and provide solutions for each set of attack.
- Hasan et al. (Tahir and Shah 2008) analyze sensor networks from a security perspective by pointing out vulnerabilities in, and give a summary of, the threat models and security benchmarks.
- Xiangqian et al. (Chen, Makki, and Yen 2009) identify various threats and vulnerabilities for WSNs and give a summary of the defence mechanisms based on networking protocol layer.

- Deva et al. (Sarma and Kar 2006) identify different types of security threats possible for a sensor network setting.

Various security attacks on WSNs and WMSNs have been classified in Figure 2.7

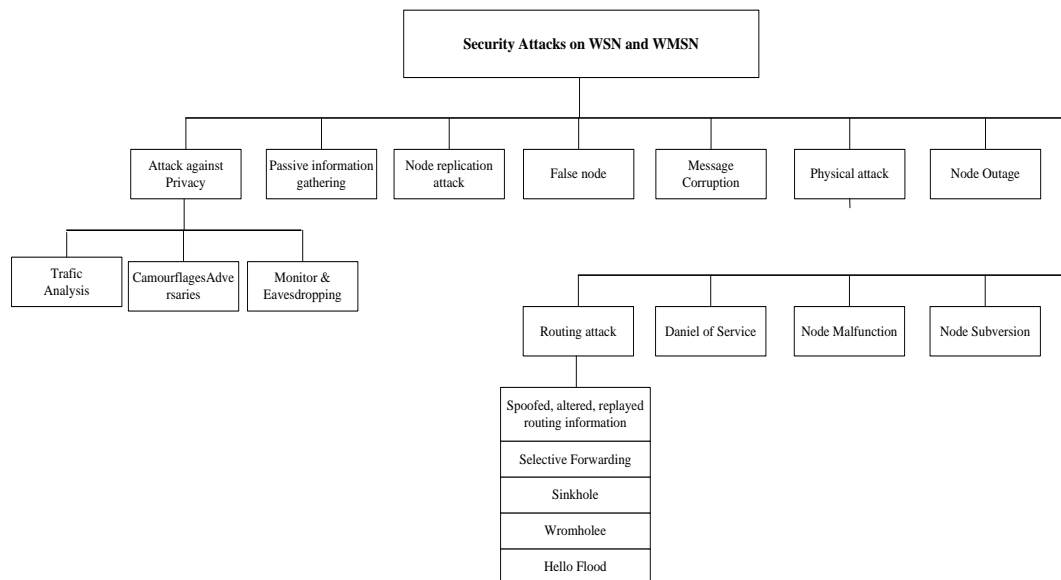


Figure 2.7 Classification of security attacks on WSNs and WMSNs (Lingxuan and Evans 2003)

In WMSNs, the multimedia sensor nodes often have to be deployed in inaccessible or hostile areas. As such, they also have the risk of being physically attacked. Some of the security solutions available for WSNs can also be adapted for use in WMSNs. However, with WMSNs, some otehr considerations also come into play, e.g. higher computation abilities and presence of video cameras on some or all of their nodes (Manel Guerrero-Zapata 2009). Therefore, the issue of ensuring security in WMSNs acquires greater complexity. Moreover, the solutions chosen have to be application oriented. Additionally, they will also depend on environmental conditions. For example, in case of video surveillance and monitoring, all the initially trusted multimedia nodes can later on be compromised and become susceptible to DoS attacks. Compromised nodes may give rise to several other problems, e.g. resistance to traffic analysis.

Many researchers have identified different types of attacks faced by both WSNs and WMSNs, such as:

- Xing Kai et al. (Xing, Srinivasan, and Li 2010) present a review of possible attacks and counter measures for WSNs based on the layering model of open system international (OSI).

The physical layer is susceptible to three types of attacks, i.e., jamming, eavesdropping, and device tampering. Traffic manipulation and spoofing are the two types of attack on the MAC layer. On the network layer, six types of attack can be launched. i.e false routing, packet replication, black hole, sinkhole, selective forwarding, and wormhole. On the application layer, there are three kinds of attack, data aggregation distortion, selective forwarding, and clock skewing. This review of possible attacks and solutions can also be used to detect the type of attack in WMSNs (Xing, Srinivasan, and Li 2010) (Ren and Yu 2006).

- Manel et al. (Manel Guerrero-Zapata 2009) give a review, and carry out an analysis of, issues related to security. which a WMSN platform design and protocol need to take into account. According to them, there is no strict boundary between WSNs and WMSNs as far as security is concerned. However, because of higher computation abilities of, and presence of video cameras on, some of the nodes, WMSNs do present some new security challenges as well as some new protection opportunities.

Some of the security challenges in WMSNs are as follows:

- As WMSNs support snapshots and audio and video streaming, the digital streams used distinguish their signals from other regular messages.
- Images, audio and video typically contain more sensitive information than scalar data. Therefore, enhancement of privacy through mechanisms, like source location, hiding, and distributed visual secret-sharing, becomes more crucial in WMSNs.
- It is possible to reduce the amount of data in WMSNs using multimedia in-network processing and compression technique.

On the other hand, some of the opportunities in WMSNs are as follows:

- The powerful WMSN sensor nodes can easily meet the demands of computation and communication required to deal with multimedia data.
 - WMSN sensor nodes have larger storage spaces than WSN nodes. This additional storage ability can be used to develop new security solutions that would not be possible in WSNs.
-

-
- The video cameras on WMSNs give them a unique advantage of detecting and identifying the attackers.

Manel Guerrero-Zapata (2009) sums up possible attacks on WMSNs as eavesdropping, sink location and location detection, compromised nodes, collusion attacks, tampering attacks and jamming attacks. Pointing at the direction of future research in WMSN security, Grieco et al. (Grieco, Boggia, and Sicari 2009) state that the research on secure WMSNs depends on how the application in question satisfies the requirements of quality, privacy and security of the data being transmitted, in addition to that of energy consumption. Privacy and authentication in WMSNs are not compartmentalized. Attacks on privacy exploiting different vulnerabilities can broadly be divided into two macro-types, eavesdropping and masquerading. Eavesdropping means sniffing the data and discovering the message being exchanged over the network. Masquerading means misrouting the data packets and retrieving the data by some malicious nodes that have masked their real nature. According to Lian et al. (Lian, Kanellopoulos, and Ruffo 2009), security issues arise because of the sensitivities of the information being transmitted in multimedia communication. With respect to the information system's complexity, there are still more attacks, e.g. eavesdropping (surreptitiously getting hold of private communication), intrusion (undesired attempt to access data), forgery (making or adapting objects with the intention of deceiving), piracy (unauthorized use of material protected under copyright law) and manipulating the computing system.

Based on the information given above, it can be concluded that WMSNs based on the OSI model are vulnerable to a variety of threats, as shown in

Table 2.1:

Table 2.1 Various types of attacks in WSN and WMSNs

Layer	Type of attacks
Application layer	Eavesdropping Masquerading Repudiation Modification Impersonation Data aggregation distortion Clock skewing Selective forwarding
Networks layer	False routing Packet replication Black hole Sinkhole Selective forwarding Wormhole Neglect and greed,
MAC layer	Traffic manipulation Identify spoofing
Physical layer	Jamming Device tempering Eavesdropping Forgery Privacy

However, we will only consider attacks at the network layer as we are basically concerned with attacks that take place in the communication channel between the processes of watermark embedding and watermark detection. WMSNs involve collaboration of sensor nodes and multimedia nodes for routing, which, however, is vulnerable to malicious attacks, like manipulation. Attackers can access the routing path to redirect the communication or broadcast false information, thus misleading the routing direction, acting as a black hole to swallow all the received messages, selectively forwarding data packets, and so on. We discuss attacks on the network layer below:

2.3.2.1 Altered Routing Information

This is the most pointed attack against a routing protocol, targeting the routing information itself at the stage of routing information exchange between the multimedia nodes. The attackers may create routing loops, extend or shorten the route, generate fake error information, partition the network, or increase end-to-end delay latency (Karlof 2003).

2.3.2.2 Selective Forwarding

Selective forwarding occurs when some malicious nodes do not forward all the messages. While some messages are forwarded, some others drop, and the dropped messages do not propagate any further in the network. This attack is possible only when the attacker lies directly on the route of data packets in the network (Lorincz, Malan, and Fulford-Jones 2004). It is of two types, selective message forwarding and selective *multimedia node* forwarding. In the former, the attacker forwards selected information from a particular node, while in the latter, the attacker drops all the information from selected nodes. The former is considered an application layer attack, while the latter is a network layer attack.

2.3.2.3 Sinkhole

In this attack, an attacker tries to take control of the whole traffic from an area by compromising a node and creating a sinkhole, with the attacker in the center. Because of either imagined or real high quality route via the compromised node, all the neighbouring nodes of the compromised node forwards data packets to the base station through it. Since all data packets are destined to the same base station, all a compromised node has to do to influence a large number of nodes is to provide a single high quality route to the base station. This makes selective forwarding rather easy, as all the traffic from the area now flows through the compromised node controlled by the attacker.

2.3.2.4 Sybil Attack

In this attack, a single node presents many identities to other nodes in the network (Karlof 2003). It can easily affect some protocols and algorithms, like fault-tolerant schemes, distributed storage, and network-topology maintenance, as, in this attack, the attacker can be in more than place at the same time.

2.3.2.5 Packet Replication

In this attack, attackers replicate some packets received previously from other nodes, and then broadcast them through the entire network even when the sender has not sent any new packets (Karlof 2003). This makes the network relay a large number of packets needlessly, causing unnecessary strain on both the power of the nodes and the bandwidth of the network. This ultimately causes network operations to terminate prematurely.

2.3.2.6 Wormhole

This attack is carried out by two or more attackers (Karlof 2003) with better communication resources than normal multimedia nodes in terms of power and bandwidth, consequently establishing a better communication channel between them. Thus, contrary to many other attacks on the network layer, this attack involves real channel. When the channel is powerful enough, other nodes end up incorporating it in their communication path, laying open their data for the scrutiny of the attackers.

This section explained the security requirements of both WSNs and WMSNs along with different types of attacks possible. The next section discusses digital watermarking process along with security requirements for, and types of attack on, this technique.

2.4 Digital Watermarking Technique

This section introduces the digital watermarking technique. The digital watermarking technique is the process of embedding a watermark into a digital media, such as image, or audio or video file, to identify the file's copyright information (Wang, Xu and Yang, 2009). Digital watermarking can be classified on the basis of different factors, like the embedding and detecting processes, the availability of the watermark signal for extraction, or the techniques used to modify the data to encode the watermark. Table 2.2 gives a detailed classification of digital watermarking techniques.

Table 2.2 Classification of digital watermarking techniques

Classification	Properties
Embedding and detecting	Spatial, Discrete Cosinus Transform (DCT) and Discrete Wavelet Transform (DWT)
Availability of the watermark signal	Blind and non-blind
Techniques used to modify data	Additive embedding Data and image fusion
Applications for <ul style="list-style-type: none"> a. Robust watermarking b. Fragile watermarking 	Broadcast monitoring, Copyright protection, Digital fingerprinting and copy control. Content authentication and integrity, Content

	archiving, Tamper detection
Cover medium	Packet data, signal, binary stream, image

Before carrying out a survey and evaluation of digital watermarking techniques applicable to both WSNs and WMSNs, it would be pertinent to explain some preliminary concepts pertaining to digital watermarking technique. This explanation will cover:

- classification of watermarks
- watermark embedding
- watermark extraction
- watermark detection

2.4.1 Classification of Watermarks

A watermark can be considered as a kind of signal or pattern inserted into a cover medium, such as binary stream, watermark bit, watermark constraint, text, image, signal or video. A watermark usually carries copyright information of the file.

Digital watermarks can be divided into two groups – *visible and invisible*. Figure 2.8 gives the classification of digital watermarks.

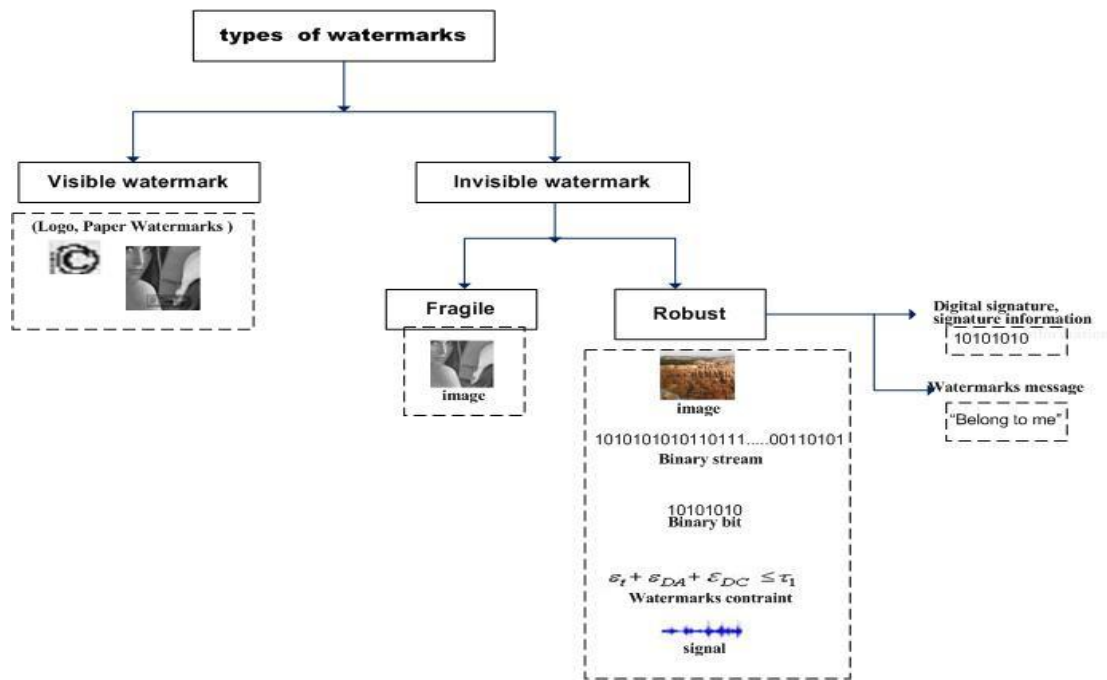


Figure 2.8 Classification of digital watermarks

A visible watermark consists of a text or image which overlies the cover medium and can be seen, but is semi-transparent, thus allowing the cover medium also to be seen. It marks the cover medium as the property of the owner of the watermark, thus providing copyright protection to it. Visible watermarks, such as logos and paper watermarks, especially when overlying the whole cover medium, are difficult to remove or tamper with, and are, therefore, robust. This makes them preferable when a digital content needs strong intellectual property protection.

An invisible watermark, on the other hand, is embedded in the cover medium and, therefore, cannot normally be seen. Information contained in these watermarks can only be extracted by electronic devices, in order to identify or prove ownership. These watermarks are used to authenticate specialized digital content which may be text, images, audio or video. Invisible watermarks can be either fragile or robust. A fragile watermark is used to verify whether the protected object was tampered with or not. This type of watermark is designed to be as fragile as possible, so that even the slightest modification of the cover medium destroys it, indicating that the cover medium was tampered with. On the other hand, a robust watermark is used to resist non-malicious distortions. These distortions usually include common image processing, geometrical transforms, and image compression.

2.4.2 Watermark Embedding Strategies

As mentioned in Section 1.4.2.2, watermark embedding is the process of combining the cover medium, the watermark signal, the sensed data and the embedding key to produce the watermarked cover medium. There are two different strategies for watermark embedding – linear additive embedding, and addition of watermark constraints into a nonlinear system equation.

The linear additive embedding strategy embeds a watermark signal by introducing linear modification in the cover medium. Here, the watermarking algorithm embeds a binary stream or binary bit of length N by modifying a subset of the cover medium coefficients. The general formula used for additive embedding is based on that given by Cox, Kilian, and Leighton (1997), as follows:

$$C_w = C_o + \alpha W_i \quad (2.10)$$

where α is the scaling parameter used to control the strength of the embedded watermark (Cox, Kilian, and Leighton 1997), C_o is the set of cover medium coefficient, W_i is the coefficient watermark, and C_w is the set of watermarked cover medium coefficient. An alternative formula given by is:

$$C_w = C_o (1 + \alpha W_i) \quad (2.11)$$

The benefit of equation 2.8 is that the coefficient C_w is small, the watermark energy is also small, and that when C_w is large, the watermark energy increases for robustness.

This thesis will use the second approach, i.e., embedding watermark constraints into the NLSP, based on the work of Koushanfar and Potkonjak (2007). NLSP, used as the cover medium, is shown in Figure 2.9, while Figure 2.10 shows an example of watermark constraints that are to be embedded into the cover medium.

$$\begin{aligned}
& \min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3 \\
& \text{Constraints} \\
& \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1 \\
& \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2 \\
& \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3
\end{aligned}$$

$C_o \approx$

Figure 2.9 NLSP used as the cover medium

$$\begin{aligned}
& \epsilon_t + \epsilon_{DA} + \epsilon_{DC} \leq \tau_1 \\
& \epsilon_t + \delta_2 \leq \tau_2
\end{aligned}$$

$W_i \approx$

Figure 2.10 Watermark constraints

$$C_w = C_o + W_i \approx$$

$$\begin{aligned}
& \text{Objective function:} \\
& \min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3 \\
& \text{Constraints} \\
& \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1 \\
& \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2 \\
& \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3 \\
& \epsilon_t + \epsilon_{DA} + \epsilon_{DC} \leq \tau_1 \\
& \epsilon_t + \delta_2 \leq \tau_2
\end{aligned}$$

Figure 2.11 Watermark constraints added to the cover medium

- where the parameters of the objective function $\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2$ and δ_3 and the parameters of the constraints $x_A, x_B, x_C, x_D, y_A, y_B, y_C, y_D, t_{DA}, t_{DB}, t_{DC}$

d_{DA}, d_{DB}, d_{DC} can be seen in subsection 2.2.1, and τ_1 and τ_2 are the values of the feasibility of the solution space. C_o is the set of the cover medium, W_i is the watermark constraint and C_w is the watermarked cover medium.

NLSP consists of the objective function used to find the value of a minimum of three non-linear constraints. The watermark constraints are added to the cover medium as shown in Fig 2.11.

2.4.3 Watermark Extraction Strategies

Watermark extraction is a process of decoding (or extracting) the watermark from the watermarked cover medium for the objective of identifying copyright (A. Herner, S. March, and J. Park 2004). There are three different approaches to decode the watermark from the watermarked cover medium – informed approach, semi-blind approach and blind approach.

The informed approach requires the cover medium in order to extract the watermark signal, whereas in the blind approach, a secret key is used to identify the location and extract the watermark. The semi-blind approach does not require the cover medium, but needs some additional information used during embedding. A watermark embedded using the linear additive embedding technique can be extracted using the cover medium. The formula to extract the watermark from the watermarked cover medium is as follows:

$$W_i = \frac{C_w - C_o}{\alpha} \quad (2.12)$$

where W_i' is the extracted watermark sequence and C_w is the possibly altered watermark image coefficient. Based on equation 2.6, we can recover the watermark sequence from the watermarked cover medium.

Watermarks with additional watermark constraints embedded in a non-linear system equation can be extracted using the cover medium. The formula to extract the watermark from the watermarked cover medium is as follows:

$$W_i' = C_w - C_o \quad (2.13)$$

where W_i' is the extracted watermark constraint and C_w is the possibly altered watermark constraint. Based on equation 2.7, we can recover the watermark constraints from the watermarked cover medium.

2.4.4 Watermark Detecting Strategies

Watermark detection is the process of measuring the correlation of the extracted watermark with the original watermark. If this correlation is higher than a specific threshold, the watermark is considered to be present, otherwise not (A. Herner, S. March, and J. Park 2004). Here, we use the objective detection approach using correlation calculation for detecting the watermark's presence.

In the objective detection approach, particularly in case of non-blind watermarking, the original watermark sequence W_i and the extracted watermark sequence W_i' are compared in order to detect the watermark's presence. The formula to calculate normalized correlation is given as:

$$NC(W_i, W_i') = \frac{W_i \cdot W_i'}{\|W_i\| \|W_i'\|} \quad (2.14)$$

Correlation can also be calculated by similarity measurements given as follows

$$SM(W_i, W_i') = \frac{W_i \cdot W_i'}{\sqrt{W_i \cdot W_i'}} \quad (2.15)$$

The value of NC varies between [-1, 1]. A value which is greater than 0 and close to 1 strongly indicates the watermark's presence; however, a lower value indicates the watermark's absence.

The detection approach using correlation calculation can also be implemented in case of additional watermark constraints embedded in the cover medium. Let X be the error of the optimal solution C_o and X' the error of the optimal solution C_w . Then, X and X' are compared to detect the watermark constraints' presence. The error of X is a set of $\{\varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \delta_3\}$ and the errors of X' is a set of $\{\varepsilon_t', \varepsilon_{DA}', \varepsilon_{DB}', \varepsilon_{DC}', \delta_1', \delta_2', \delta_3'\}$. The formula to calculate similarity is given as:

$$SM(X, X') = \frac{|\varepsilon_t + \varepsilon_{DA} + \varepsilon_{DB} + \varepsilon_{DC} + \delta_1 + \delta_2 + \delta_3| \cdot |\varepsilon'_t + \varepsilon'_{DA} + \varepsilon'_{DB} + \varepsilon'_{DC} + \delta'_1 + \delta'_2 + \delta'_3|}{\sqrt{|\varepsilon_t + \varepsilon_{DA} + \varepsilon_{DB} + \varepsilon_{DC} + \delta_1 + \delta_2 + \delta_3| \cdot |\varepsilon'_t + \varepsilon'_{DA} + \varepsilon'_{DB} + \varepsilon'_{DC} + \delta'_1 + \delta'_2 + \delta'_3|}} \quad (2.16)$$

The value of similarity varies between [-1, 1].

This concludes the discussion of watermark classification and watermarks embedding, extraction and detection processes. The next section explains the security requirements for digital watermarking technique.

2.4.5 Security Requirements for Digital Watermarking Technique

Most digital watermarking applications have particular security requirements of their own and no general set of requirements can be given that would apply to all digital watermarking techniques. Yet, some general indications can be made for a large number of applications. For example, one fundamental requirement is imperceptibility which applies to all applications irrespective of their purposes. These requirements have been discussed below.

2.4.5.1 Imperceptibility

As mentioned above, imperceptibility, or perceptual transparency of the watermark, is a fundamental requirement for all digital watermarking applications. This means that the embedded watermark should not degrade the quality of the cover medium to any significant degree and should not interfere with the user's interaction with it. A digital watermark is imperceptible if watermarked cover medium cannot be distinguished from the original cover medium by human users. On comparing the watermarked cover medium with the original cover medium, slight modifications may be apparent. However, normally, the users can't compare the two as the original cover medium is not accessible to them. Therefore, it is often sufficient for fulfilment of the requirement of imperceptibility that the modifications in the watermarked cover medium, if any, are not noticed by the users as long as they do not have the opportunity to compare it with the original cover medium.

2.4.5.2 Robustness

Robustness of a watermark is its ability to resist any intentional alteration or removal by either standard or malicious processing or attacks. The standard processing usually includes common image processing, geometrical transformation, and image compression. The common image processing operations include noise insertion, contrast adjustment, and cropping. The geometrical transforms include rotation, scaling and translation. A watermark is said to be robust against compression if it can be detected even after the image is compressed. Among the standard processing operations mentioned, geometric transformation can be undertaken by using an image processing software and defeating the purpose of the watermark by making it undetectable. A watermark is termed robust if it can resist such intentional alterations or removal. If it cannot, it is termed fragile. A robust watermark is designed to resist normal processing caused during common image processing operations. Fragile watermarks, on the other hand, are easily destroyed by slight distortions.

2.4.5.3 Blind or Informed Detection

In some digital watermarking applications, like copyright protection and data monitoring, the original, un-watermarked cover medium is available during the detection process. In other digital watermarking applications, such as copy protection and indexing, the detection process has to be carried out without the original cover medium. The detection process in the former watermarking system is known as informed detection while, in the latter case, it is blind detection. The literature on digital watermarking systems also calls informed detection private watermarking system, and blind detection public watermarking system. Fragile watermarking certainly requires public watermarking, but it also increases the chances of malicious attacks.

2.4.5.4 Security

A secure digital watermarking technique can survive many hostile attacks that try to defeat the purpose of the watermark. These attacks include unauthorized removal, modification, manipulation or detection of the watermark. Security in digital watermarking can be compared to security in encryption techniques. The most direct approach to make the watermark information secure is to apply Kirchhoff's Principle in cryptography (Massey 1992). This principle assumes that the data encryption method used is known to unauthorized parties and, therefore, security lies in the choice

of a key. This means that, to ensure secrecy, it is imperative to use a secret key in the process of embedding and detection. Thus, a watermark will really be secure only if unauthorized parties cannot detect the watermark's presence, nor remove it, even if they know the exact algorithm for embedding and extracting the watermark.

This concludes the discussion of security requirements for digital watermarking technique. The next section elaborates different kinds of attacks on digital watermarking technique.

2.4.6 Attacks on Digital Watermarking Techniques

Potentially, digital watermarking encompasses a number of applications, including copyright protection, broadcast monitoring, authenticating content and ensuring its integrity, and providing authorized access and covert communication control. However, a watermark embedded in a cover medium may become vulnerable, or may be unintentionally impaired during processing. Sometimes processing may intentionally aim at obstructing watermark reception. In digital watermarking, any processing that impairs watermark detection or communication of watermark information is known as attack (Shrekar 2011, Voloshynovskiy, Pun, and Eggers 2001).

To serve its purpose, a digital watermarking technique must be robust against these attacks. A watermark is called robust if, even after an attack on the data, it is not impaired in any way. Criteria like bit error probability or channel capacity can be used to measure impairment of a watermark. A simultaneous consideration of watermark impairment and data distortion as a result of the attack can be used to evaluate the robustness of a digital watermarking technique.

Following is a classification of a wide range of currently known attacks, followed by a discussion of each. Broadly, these attacks can be classified into removal, geometric, cryptographic, and protocol attacks (Shrekar 2011, Voloshynovskiy, Pun, and Eggers 2001), as shown in Figure 2.12.

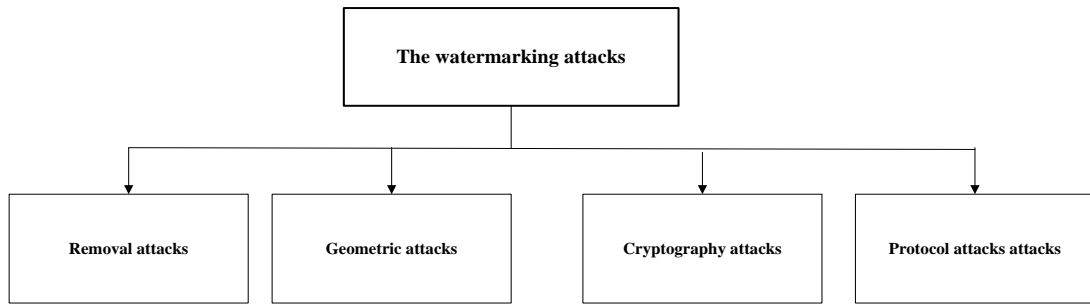


Figure 2.12 A broad classification of currently known attacks on digital watermarking techniques

A discussion on each follows:

2.4.6.1 Removal Attack

This attack aims at completely removing the watermark from the watermarked cover medium, so that no amount of processing, howsoever complex, can recover it from the attacked cover medium. This attack is not used to crack the security algorithm, like the watermark embedding key. It includes processes like quantization, demodulation, denoising, and collusion attacks. Not all these processes are always able to completely remove the watermark, yet they may cause irraparable damage to the watermark information. For example, in collusion attack, illustrated in Figure 2.13, an attacker or group of attackers may obtain several copies of a given data set, each of which will be signed with a different key or watermark, and then average all the copies or use small parts of each copy, to achieve their objective.

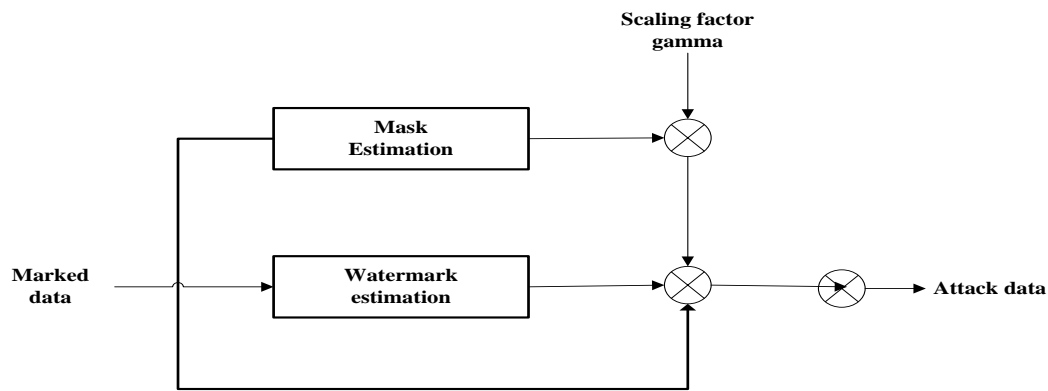


Figure 2.13 A perceptual remodulating attack

2.4.6.2 Geometric Attack

Instead of removing the watermark, this attack aims at distorting the synchronization of the watermark detector with the embedded information. Unintentional geometric attacks include image-processing manipulations, such as scaling images, printing and scanning marked documents, changing a digital video's aspect ratio, and cropping an image to extract a region of interest (Licks and Jordan 2005). Every geometric attack is defined by a set of parameters that determines the operation performed over the target image, for example, a rotation angle applied to the sampling grid characterizes a rotation attack (Licks and Jordan 2005).

However, on regaining perfect synchronization, the watermark detector can recover the embedded information. Further, most newer digital watermarking techniques use special synchronization techniques to thwart these attacks. An invariant transform domain or an additional template, enabling the geometric distortion to be estimated, is used to make digital watermarking robust against general geometric distortions (Shrekar 2011, Voloshynovskiy, Pun, and Eggers 2001).

2.4.6.3 Cryptographic Attack

This type of attack attempts to crack the security algorithm in the digital watermarking technique used, and thus find a way of either removing the embedded information or embedding misleading information, which is usually computationally a highly complex watermark. For example, the brute-force search technique is one of the techniques used to crack the embedded secret information. On the other hand, Oracle attack is another technique used to create non-watermarked signals, if a watermark detector is available. Practically, however, the Oracle attack has only a

limited scope to be used because it is computationally very complex (Sherekar 2011, Voloshynovskiy, Pun, and Eggers 2001).

2.4.6.4 Protocol Attack

This type of attack aims to hijack the whole concept of watermarked application. Copy attack is one such attack which aims at estimating a watermark from the watermarked cover medium and then copying it to some other cover medium, rather than destroying it. The attackers adapt the estimated watermark to the local features of the targeted recipients to fulfil the requirement of imperceptibility. The copy attack can succeed only when it is possible to embed a valid watermark in the targeted cover medium without the knowledge of either the watermark key or the algorithm the watermarking technology has used. Depending on signals, watermarks may resist copy attacks, or the attacker may obtain different watermarks. For example, an attack can be successfully carried out by averaging all the different copies (Sherekar 2011, Voloshynovskiy, Pun, and Eggers 2001).

This section discussed the digital watermarking process along with their security requirements and types of possible attacks. The next section specifically discusses the security requirements of digital watermarking for both WSNs and WMSNs with different types of attacks.

2.4.7 Working of Digital Watermarking Technique in WSNs

This section explains how digital watermarking technique can be used for WSNs' security. The different ways to do so are implanting the specific name or message of the owner for copyright data protection in WSNs. To accomplish this, a usual encoder in a WSN requires: the original sensed data d_o , the watermark message w and the watermark key k . The watermark message w and watermark key k are inserted into the watermark generator to produce the watermark signal w' . The input parameters mentioned above are taken by an embedding function F which gives the watermarked data $d_w = F_{embed}(d_o, w', k)$ as output. This watermark key k is used for both embedding and detection process. If the key is not used, the watermarking algorithms have to be kept secret, but this is normally not the case. In the process of detection, the input parameters for the watermarking detector are: the watermarked data d_w , the original sensed data d_o , the watermark signal w , and the watermark key k , and the watermark detection process is defined as

$w_e = F_{detect}(d_w, k, w', d_o)$. The process of extraction does not need the original sensed data d_o and the watermark w because the detection algorithm can usually retrieve the hidden information even if the original signal is not known. The procedure gives the estimated watermark w_e as the output. In the process of extraction, the watermark w_e is extracted by using the watermark key k to get the watermark signal w' and the sensed data d .

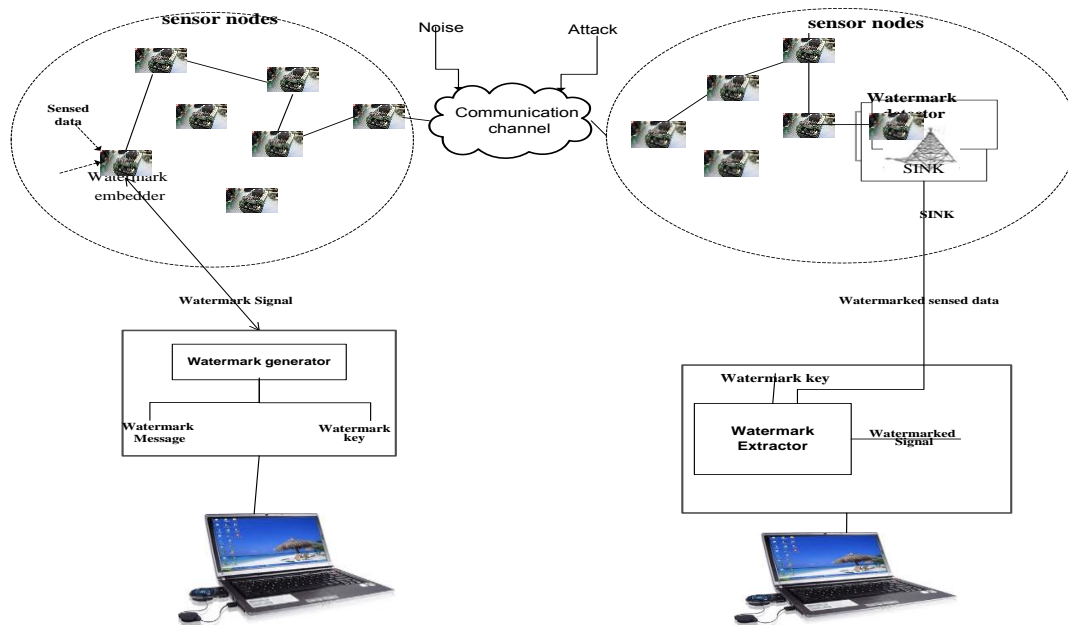


Figure 2.14 Working of digital watermarking technique in WSNs

As described in Section 1.5, there is a very pressing need to address the problem of copyright protection of data in WSNs. Digital watermarking technique can be used for the purpose, as shown in Figure 2.14.

With this explanation of the working of digital watermarking technique in WSNs, the next section moves to explaining the working of digital watermarking technique in WMSNs.

2.4.8 Working of Digital Watermarking Technique in WMSNs

This section explains how WMSNs and digital watermarking technique can work together. It will show that the technique can be used to implant the specific name or information of the owner in WMSNs. For the purpose, a usual encoder for WMSNs requires: the original image i obtained by the video sensor, which is then sent to the multimedia sensor node. The WMSN is managed by a

laptop that captures this image. Now, the watermark message is inserted into it. The process of embedding is as follows: first the image is decomposed into several bands, then the pseudo-random sequences are added to the larger coefficients which are not located in the lowest resolution. The DWT watermark inserted algorithm consists of four parts, namely original image, calculation of multilevel threshold, watermark embedding process, and inverse discrete wavelet transform (IDWT). The watermarked image obtained after the embedding process is sent to the multimedia sensor node by a laptop. This image then transmits through a communication channel to a sink. The watermarked image is again managed by the user with the laptop. The process of detecting or extracting is the inverse procedure of the watermark insertion process. It requires the watermarked image and the key.

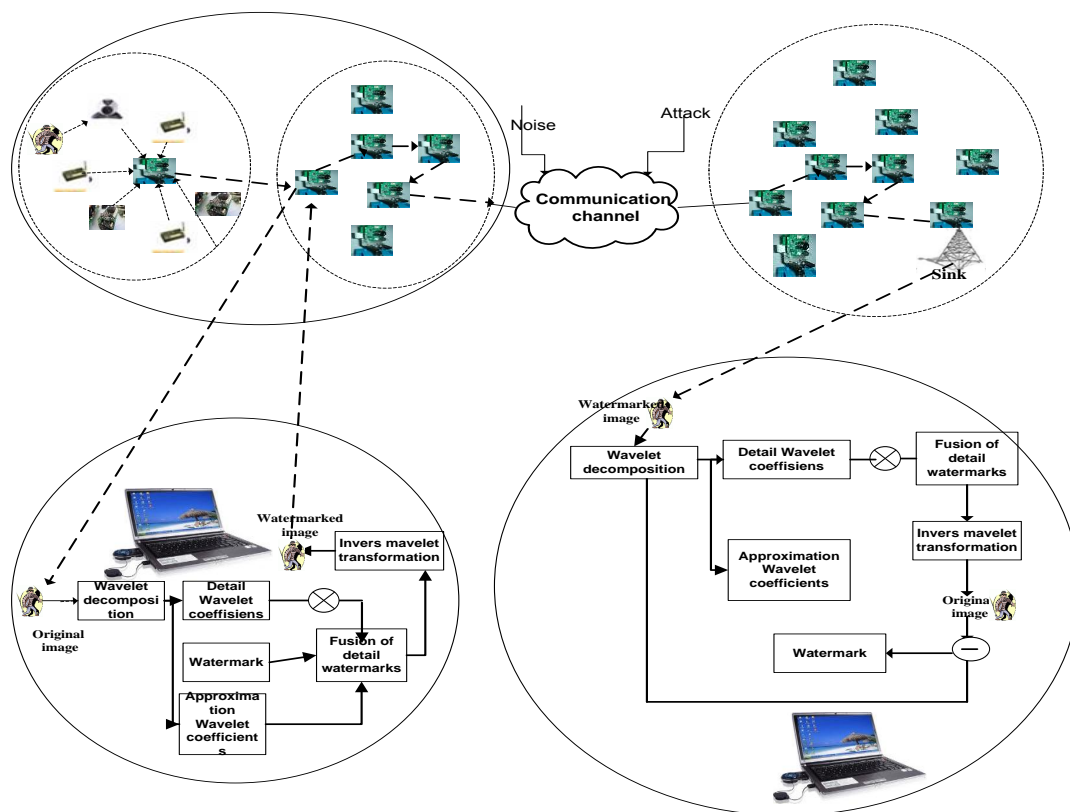


Figure 2.15 Working of digital watermarking technique in WMSNs

As described in Section 1.6, there is a very pressing need to address the problem of content ownership detection in WMSNs. This cannot be done by using cryptography, because cryptography protects information only during communication and cannot prove content ownership. Thus we need to study the possibility of using watermarking technique to address the problem. Digital watermarking technique and WMSNs can work together, as shown in Figure 2.15.

This section discussed how digital watermarking technique and WMSNs can work together. The next section will explain the evaluation framework for digital watermarking techniques in WSNs.

2.5 Evaluation Framework for Digital Watermarking Techniques in WSNs

This section carries out a survey of the available literature on WSNs and digital watermarking technique. To adequately analyse the algorithms proposed in the existing literature, we evaluate all the WSN digital watermarking algorithms across 10 parameters. These include: cover medium, watermark message, sensed data, type of watermark signal, watermark key, watermark generator, watermark embedding technique, watermark detection technique, type of detection, noise, and attacks. Figure 2.16 shows these 10 parameters for evaluation of digital watermarking techniques in WSNs. The main idea behind adopting this approach is to ensure an independent and thorough evaluation of each algorithm, by clearly studying each aspect independently. In all, 11 different algorithms will be evaluated on these 10 parameters. Table 2.2 gives comprehensive details of these 11 algorithms.

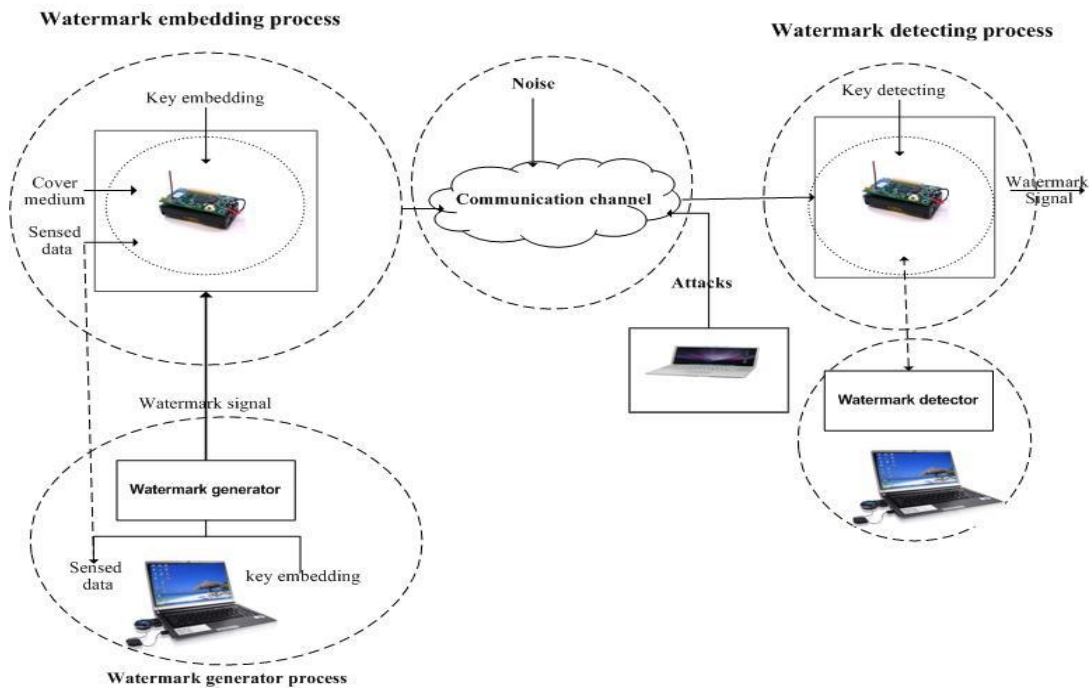


Figure 2.16 The 10 parameters for evaluation of digital watermarking in WSNs

This section presented the 10 parameters used for evaluation of digital watermarking in WSNs. The next section will carry out an in-depth survey of the literature on digital watermarking in WSNs.

2.6 Survey of Literature on Digital Watermarking Technique in WSNs

This section evaluates each of the 11 algorithms mentioned above on each of the 10 parameters individually. The purpose is to identify the similarity and differences among these 11 algorithms and try to understand the rationale behind the authors' selection of a particular value for each parameter in their procedural solution. We begin the discussion with cover medium.

2.6.1 Cover Medium

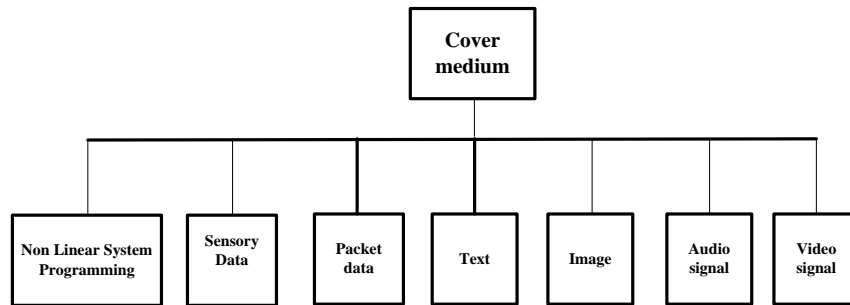


Figure 2.17 Cover medium

Cover medium is one of the key components of watermarking technique. Examples of cover medium include sensory data, packet data, texts, images, audio signals, video signals, and the Non Linear Programming System (NLSP), as shown in Figure 2.17. Among all these cover media, NLSP and sensory data are slightly different. While all other media are covers in themselves, NLSP is the process of generating a cover medium and then using it. Similarly, the sensory data consists of sensor nodes that capture the data. These sensor nodes are deployed in 2-D coordinates and use this sensory data.

NLSP is produced by using the atomic trilateration process. This is a process by which the position of a sensor node can be determined, using the positions of, and its distances from, at least three other sensor nodes in the WSN. The watermark constraints are embedded during this process (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007).

Following is a comparative evaluation of different cover media used in watermarking techniques in WSNs. All these different approaches have been presented in

Table 2.1. Most of these use ‘packet data’ as the cover medium (Radu, Mikhail, and Sunil 2004), (Xiangqian 2009), (Xuejun 2010), (Rong, Xingming, and Ying 2008), (Xiaomei and Xiaohua 2009), (Juma, Kamel, and Kaya 2008, Ren 2011). One of these has used sensory data (Zhang, Liu, and Das 2008) while two have used NLSP (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007).

Packet data have been found to be better than sensory data by us as they form the basic unit of communication over digital networks. The packet data are parts of the whole data. For transmission, the whole data are first broken down into smaller chunks, called packets

(Wuyungerile, Bandai, and Watanabe 2010). The advantage of using packet data is that, with them, the sensor nodes consume network resources only when data are actually being transferred. This makes the data transfer process suitable to wireless sensor nodes as these nodes work under severe energy and resource constraints.

Table 2.3 Cover media used in literature

Author	Year	Cover medium
Feng et al.	2003	Non Linear Programming System
Sion et al.	2004	Packet data
Koushanfar et al.	2007	Non Linear Programming System
Albath et al.	2007	Packet data
Zhang et al.	2008	Sensory data
Juma et al.	2008	Packet data
Xiao et al.	2008	Packet data
Xiaomei et al.	2009	Packet data
Xuejun	2010	Packet data
Wang et al.	2011	Packet data
Kamel et al.	2011	Packet data

This thesis uses NLSP as cover medium for its algorithm, because the type of watermark used in it can only be embedded in this cover medium. The watermark used will be explained in Section 2.6.4.

2.6.2 Watermark Message

Watermark message can be a simple text or an image depending upon the application requirements. In watermarking process for WSN applications, most often plain texts are used, because using images as watermarks will strain the network resources too much. Examples of some possible watermark messages have been given in Figure 2.18.

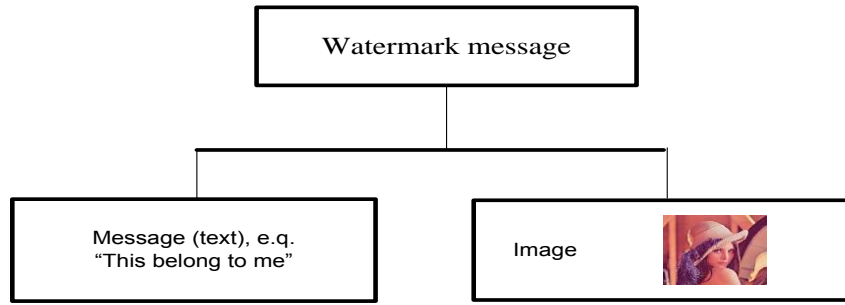


Figure 2.18 Watermark message

The watermark message ‘this belong to me’ is encoded by MD5 and this operation results in a message digest. This digest is then encoded by RC5 and this operation results in a binary stream. This binary stream is the watermark message that is to be communicated. It is shown in Figure 2.19.

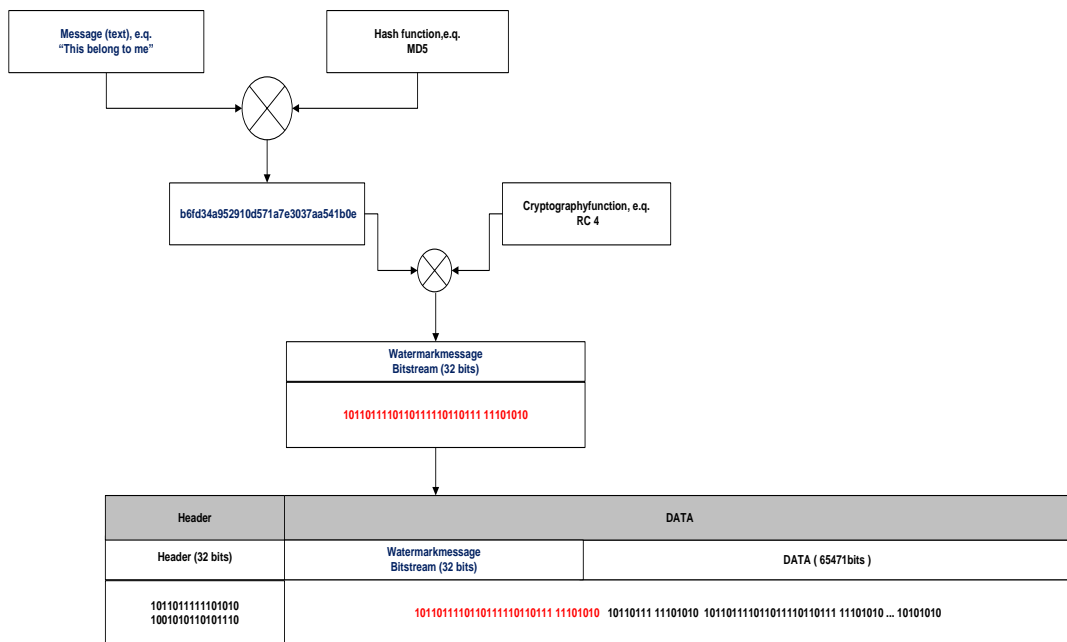


Figure 2.19 Encoded watermark message

A comparative evaluation of different kinds of messages to be protected using watermarking technique in WSNs follows next. Overall, all these different approaches have been presented in Table 2.4. Although most of the works do not mention the specific watermark message used by them (Radu, Mikhail, and Sunil 2004), (Zhang, Liu, and Das 2008), (Xuejun 2010), (Rong, Xingming, and Ying 2008), (Xiaomei and Xiaohua 2009), Fang (Jessica and Potkonjak 2003),

Koushanfar (Koushanfar and Potkonjak 2007), two specify using text as their watermark message. (Jessica and Potkonjak 2003).

Table 2.4 Watermark messages used in literature

Author	Year	Wateramrk message
Feng et al.	2003	Plain text (e.g. this belongs to me)
Sion et al.	2004	---
Koushanfar et al.	2007	Plain text (e.g. this belongs to me)
Albath et al.	2007	---
Zhang et al.	2008	---
Juma et al.	2008	---
Xiao et al.	2008	---
Xiaomei et al.	2009	---
Xuejun	2010	---
Wang et al.	2011	---
Kamel et al.	2011	---

2.6.3 Sensed Data

The term sensed data means the phenomena captured from the environment by the sensor nodes of a WSN and communicated to the sink, e.g. atmospheric temperature and humidity, wind speed and direction, light intensity, etc. These sensed data constitute the information to be secured through watermarking. The sensed data are analogue in nature and, hence, need to be converted to digital format before transmission. The digitized information is then encoded by a hash function MD5 to generate a fixed length binary stream. As shown in Figure 2.20, this stream is the sensed data to be communicated. The binary stream consists of one or more bytes of arbitrary information. The binary matrix is a matrix with entries 0 and 1 (Akyildiz, Pompili, and Melodia 2005).

A comparative evaluation of different kinds of messages to be protected using watermarking technique in WSNs follows next. Overall, all these different approaches have been presented in Table 2.5. Most of the works use ‘binary stream’ as the sensed data (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Xuejun 2010), (Rong, Xingming, and Ying 2008), (Xiaomei and

Xiaohua 2009), (Wang, Sun, and Ruan 2011). One mentions ‘binary matrix’ used as the sensed data (Zhang, Liu, and Das 2008). Others, however, make no mention of the sensed data used (Xiangqian 2009), (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007).

For our purposes, binary stream is better than binary matrix to be used as sensed data, because it is a small chunk of the whole data and, therefore, the sensor node does not need too much energy to transmit it (Wuyungerile, Bandai, and Watanabe 2010). On the other hand, binary matrix is a pseud-random code generated as a modulation pulse for the purpose of spreading signals across the entire band, and there is no mention of the amount of energy required by it (Zhang, Liu, and Das 2008).

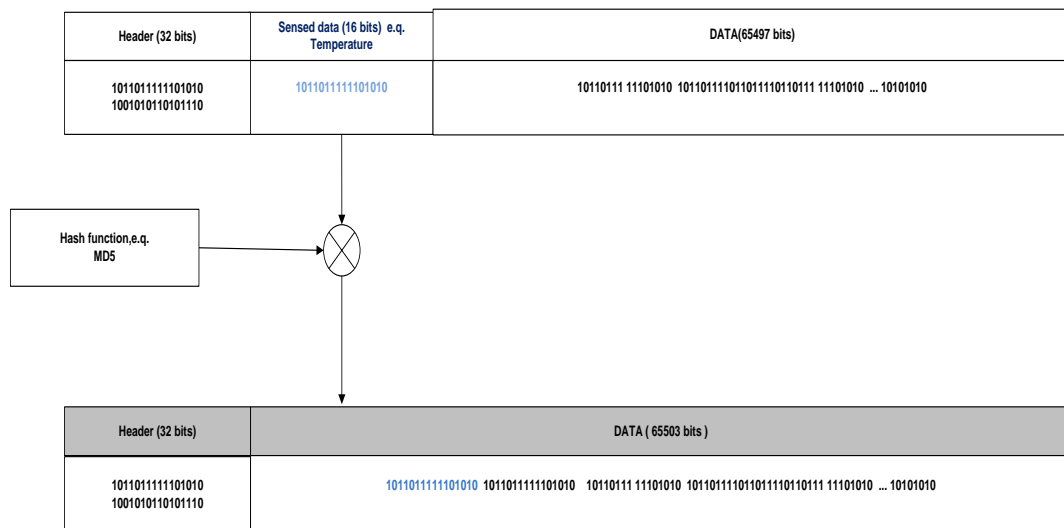


Figure 2.20 An example of sensed data encoded by hash function

Table 2.5 Sensed data used in literature

Author	Year	Sensed data
Feng et al.	2003	---
Sion et al.	2004	Binary stream (101010 ...)
Koushanfar et al.	2007	---
Albath et al.	2007	Binary stream (101010 ...)
Zhang et al.	2008	Binary matrix $\begin{bmatrix} 1 & 0 & . & . & 1 \\ 1 & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 1 & . & . & 1 \end{bmatrix}$
Juma et al.	2008	Binary stream (101010 ...)

Xiao et al.	2008	Binary stream (101010 ...)
Xiaomei et al.	2009	Binary stream (101010 ...)
Xuejun	2010	Binary stream (101010 ...)
Wang et al.	2011	Binary stream (101010 ...)
Kamel et al.	2011	Binary stream (101010 ...)

2.6.4 Type of Watermarks

A pattern of bits inserted into a cover medium used to identify illegal copies, examine the authenticity of the document, or detect tampering of digital content, is called a watermark. There are different types of watermark, such as watermark constraints, binary stream, and watermark decimal matrix. Examples of these watermarks are given in Figure 2.21

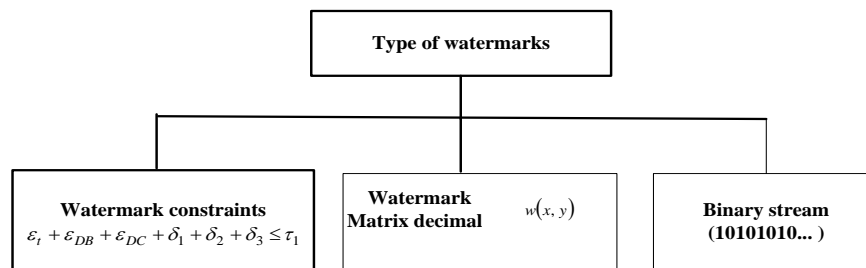


Figure 2.21 Types of watermarks used in WSN

Following is a comparative evaluation of different types of watermarks used in WSNs, as found in the existing literature. Overall, all these different approaches have been presented in Table 2.6.

In most of these approaches, ‘binary stream’ has been used as the watermark signal (8, 16, 64, 128 bits) (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Xiangqian 2009), (Xuejun 2010), (Rong, Xingming, and Ying 2008), (Xiaomei 2009, Xiaomei and Xiaohua 2009), (Juma, Kamel, and Kaya 2008), (Wang, Sun, and Ruan 2011). Two of them use watermark constraints (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007), while one suggests using a watermark decimal matrix $w(x, y)$.

Watermark binary stream proves itself better than other watermark signals because it is possible to produce it through a calculation of the watermark bit of MSB, which is the most significant bit of the hash function (Rong, Xingming, and Ying 2008), (Xiaomei and Xiaohua 2009). Additionally, it is also possible to compute it using the hash function by concatenating all the individual data

elements in the group and applying this function. Kamel and Juma (2011) and Xuejun (2010) have used the hash function to produce the binary stream. This approach uses less energy and shows greater computational efficiency than watermark bitshigh cost energy and computationally less efficient than the binary stream.

Table 2.6 Types of watermark used in literature

Author	Year	Type of watermark
Feng et al.	2003	Watermark constraint $\varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4$
Sion et al.	2004	Binary stream (101010 ...)
Koushanfar et al.	2007	Watermark constraint $\varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4$
Albath et al.	2007	Binary stream (101010 ...)
Zhang et al.	2008	Watermark decimal matrix $w(x, y) = \begin{bmatrix} 2 & 3 & . & . & 3 \\ . & . & . & . & . \\ 1 & 5 & . & . & 4 \end{bmatrix}$
Juma et al.	2008	Binary stream (101010 ...)
Xiao et al.	2008	Binary stream (101010 ...)
Xiaomei et al.	2009	Binary stream (101010 ...)
Xuejun	2010	Binary stream (101010 ...)
Wang et al.	2011	Binary stream (101010 ...)
Kamel et al.	2011	Binary stream (101010 ...)

However, even though watermark binary streams are better than watermark constraints, we have used watermark constraints in our algorithm because they can be embedded in NLSP while binary streams cannot. .

2.6.5 Watermark Key

The key K in a watermarking procedure is a means to build a parameterized set of embedding and detection method. It should not be confused with the cryptography key used to encrypt and decrypt the messages carried by the watermarking channel. It is used in both embedding and detection processes in the watermarking techniques in WSNs.

The two main watermark keys found in the literature are (1) pseud-orandom sequences, and (2) binary streams. Examples of watermark keys are shown in Figure 2.22

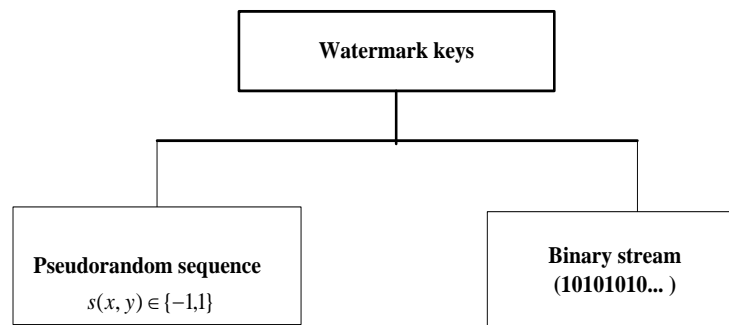


Figure 2.22 Type of watermark keys

Following is a comparative evaluation of the different types of watermark keys found in the literature. Overall, all these different approaches have been presented in Table 2.7. Most of them use ‘binary stream’ for watermark key (Kamel and Juma 2011, Xiangqian 2009), (Xuejun 2010), (Rong, Xingming, and Ying 2008), (Xiaomei 2009, Xiaomei and Xiaohua 2009). A few use pseudo-random sequence (Potdar, Sharif, and Chang 2011), while others make no mention of the watermark keys used by them (Xuejun 2010), (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007), (Radu, Mikhail, and Sunil 2004).

Genererally speaking, binary streams score over pseudo-random sequences $s(x,y)$ as watermark keys, becuae they are more energy efficient. Additionally, it is possible to easily produce a watermark signal using binary streams, by adding them to a hash function. On the other hand, pseudo-random sequences $s(x,y)$ are computationally exhaustive and less energy efficient. Additionally, a pseudorandom sequence $s(x,y)$ generated by the sink S_i $i=1,2,\dots,L$ requires the sinks to be non-overlapping with each other, spreading each single watermark bit b_i $i=1,2,\dots,L$ onto each S_i $i=1,2,\dots,L$

Table 2.7 Watermark keys used in literature

Author	Year	Watermark Key
Feng et al.	2003	---
Sion et al.	2004	---
Koushanfar et al.	2007	---

Albath et al.	2007	Binary stream (101010 ...)
Zhang et al.	2008	Pseudorandom sequence $s(x,y)$
Juma et al.	2008	Binary stream (101010 ...)
Xiao et al.	2008	Binary stream (101010 ...)
Xiaomei et al.	2009	Binary stream (101010 ...)
Xuejun	2010	---
Wang et al.	2011	---
Kamel et al.	2011	Binary stream (101010 ...)

We also use binary stream (101010 ...) as the watermark key to generate watermarks in this thesis.

2.6.6 Watermark Generator

A mathematical function used to generate watermark signals through a watermark message and watermark key is called watermark generator. Literature on watermarking in WSNs reveals two kinds of watermark generators, i.e. (1) hash function, and (2) product function.

A product results from multiplying two factors, or is an expression that identifies the factors to be multiplied. A product function consists of two or more functions that have to be multiplied.

The product function of $b_i \in \{-1,1\}$, $\alpha(x,y)$ and $s(x,y)$ produces the watermark decimal matrix $w(x,y)$ (Potdar, Sharif, and Chang 2011), where $b_i \in \{-1,1\}$ are the watermark bits to be embedded in the sensor nodes, the random value $\alpha(x,y)$ is equal to the amplitude of the watermark for the node $s(x,y)$, and $s(x,y)$ are the 2-D coordinates representing the position of the sensor node (Zhang 2008). A comparative evaluation of the different watermark generators is given below. Overall, all these different approaches have been presented in Table 2.8.

Most of the works in WSN watermarking literature use 'hash function' for watermark generation, e.g. MD5 and SHA (Kamel and Juma 2011), (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007), (Xiangqian 2009), (Ming-Kuei 1962), (Xiaomei 2009, Xiaomei and Xiaohua 2009). One of them (Zhang, Liu, and Das 2008) uses the product function of

$b_i \in \{-1,1\}, \alpha(x, y)$ and $s(x, y)$ to generate a watermark decimal matrix, while others make no mention of the watermark generator used by them (Rong, Xingming, and Ying 2008), (Juma, Kamel, and Kaya 2008).

However, being a deterministic procedure taking arbitrary blocks of data, e.g. binary stream, and returning a fixed-size bit string, hash function may be said to be better than the product function of $b_i \in \{-1,1\}, \alpha(x, y)$ and $s(x, y)$. The hash function encodes the binary stream and turns it into the hash value, also known as the message digest. Using hash function as the watermark generator has four significant advantages: 1) computing the hash value is easy for any given binary stream, 2) generating message is not feasible for a given hash, 3) modifying a message is not possible unless the hash is changed, and 4) modifying a message is not feasible unless the hash is changed. Depending on their properties, the hash function is generally found suitable as the watermark generator for WSNs (Elkamchouchi, Emarah, and Hagra 2006, Subash and Divya 2011), with another advantage of being energy efficient, while the product function of $b_i \in \{-1,1\}, \alpha(x, y)$ and $s(x, y)$ incurs high computational costs.

Table 2.8 Watermark generators used in literature

Author	Year	Watermark generator
Feng et al.	2003	Hash function (MD5), RSA, RC4
Sion et al.	2004	Hash function
Koushanfar et al.	2007	Hash function (MD5), RSA, RC4
Albath et al.	2007	Hash function
Zhang et al.	2008	Product function of $b_i, \alpha(x, y)$ and $s(x, y)$
Juma et al.	2008	Hash function
Xiao et al.	2008	Hash function
Xiaomei et al.	2009	Hash function
Xuejun	2010	---
Wang et al.	2011	Hash function
Kamel et al.	2011	Hash function

Our algorithm uses LFSR and Kolmogorov rule as watermark generators to produce watermark constraints.

2.6.7 Watermark Embedding Technique

The process of inserting watermark signal into a cover medium is known as watermark embedding. This process has already been explained in detail in Section 1.4. Below, we critically evaluate various embedding techniques for WSNs proposed in the literature. All these are summarized in Table 2.9. Most of the works use ‘Least Significant Bits (LSB)’ as embedding technique (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Albath 2007), (Zhang 2007), Juma (2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010), (Ren 2011). Two of them have used ‘MSB’ (Sion, Atallah Mikhail, and Prabhakar Sunil, 2004) (Radu, Mikhail, and Sunil 2004), Xiaomei 2009), and one has used the DSSS system, which involves the generation of a pseudo-random code that serves as the modulation pulse spreading the signals throughout the band (Zhang, Liu, and Das 2008). ‘Adding watermark constraint to the processing step during the network operation’ has also been used as an embedding technique by two (Jessica and Potkonjak 2003), (F. Koushanfar Auerbach publications 2007).

This latter approach and also the DSSS system have certain advantages over LSB and MSB. The DSSS system can survive attacks like false distribution imposition on the whole sensor, false distribution imposition on a part of the sensor, and remnant check transformation. Similarly, watermark constraints added to the processing step during the network operation can resist attacks like ghost signatures, removing the author’s signature, adding new signatures, and de-synchronization.

On the other hand, LSB and MSB have the advantages of being the simplest methods for embedding watermarks and suit the low-powered wireless sensor nodes very well. These techniques use a simple replacement procedure for embedding watermark signal, viz. replacing LSB from the cover medium.

As these methods use the whole cover medium for transmission, utilizing the high capacity of the WSN channel, they allow multiple embedding of a watermark signal. Thus, even if most signals

are removed by the attackers, some would remain to prove ownership. Even a single watermark signal is enough for the purpose.

Table 2.9 Watermark embedding techniques used in literature

Author	Year	Watermark embedding technique
Feng et al.	2003	Adding watermark constraint to processing step during network operation
Sion et al.	2004	Selection criteria using MSB
Koushanfar et al.	2007	Adding watermark constraint to processing step during network operation
Albath et al.	2007	generating the one-time pad by repeatedly concatenating the substring
Zhang et al.	2008	The waterark sensory data $d(x, y) = w(x, y) + o(x, y)$ $w(x,y)$ is the watermark for sensor node and $O(x,y)$ is sensory data
Juma et al.	2008	Concatenation of the current group hash value group g_i and next group hash value group g_{i+1} . $W = \text{HASH} ([K g_i g_{i+1}]) g_i$
Xiao et al.	2008	By modification the embedding bit of each packet. LSB
Xiaomei et al.	2009	The random value of each send data and time was calculated by inputting the collection time and its MSB and the key K into random function.
Xuejun	2010	IIS = input integer stream IBS=input binary stream T = Threshold If $IIS \geq T$ “IBS=1” become “IBS=0” else “IBS=0” become “IBS=0”
Wang et al.	2011	Embed bit of watermark by changing the parity of its LSB.
Kamel et al.	2011	Concatenation of the current group hash value group g_i and next group hash value group g_{i+1} . $W_i = \text{HASH} (K g_i SN)$ $SN = \text{serial number}$

Our algorithm adds watermark constraints to the processing step during the network operation because our technique aims at surviving attacks that modify and remove data, such as false data insertion, data modification and Sybil attack.

2.6.8 Watermark Detecting Technique

As mentioned in Section 1.4.2.4, watermarking technique consists of two processes, embedding and extraction or detection. This section explains the process of extraction or detection. This process determines whether the given cover medium contains a particular watermark or not. Following is a comparative evaluation of different detection techniques used for watermarking in WSNs as found in the literature. Overall, all these different approaches have been presented in Table 2.10.

Most of these approaches use ‘statistical correlation’ for detection of watermarks, e.g. probability, similarity, mean deviation, standard deviation, and Gaussian hypothesis (Radu, Mikhail, and Sunil 2004), (Albath and Madria 2007), (Zhang, Liu, and Das 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010). Some have used synchronization points (Juma, Kamel, and Kaya 2008), manipulation of embedding bit (Rong, Xingming, and Ying 2008), and comparison of the hash calculated between the sender and the receiver (Kamel and Juma 2011). Others do not mention the detection technique used by them (Jessica and Potkonjak 2003), (F. Koushanfar Auerbach publications 2007), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011).

In our estimation, statistical correlation is a better method for watermark detection than the manipulation of the embedding bit or the comparison of the hash calculated between the sender and the receiver, as it does not need to use the original cover medium sent from the sensor node in order to detect the watermark (Wenjun and Liu 1999). Usually, detecting the watermark in a cover medium requires first extracting the watermark signal, which normally necessitates using the original cover medium. The extracted watermark signal is then compared with the original watermark signal. Next, the similarity index between the two is compared with a threshold, which determines whether the tested cover medium is the original cover medium’s watermarked version or not (Wenjun and Liu 1999). However, in case of WSNs, the watermark signal is invisible, and so it is not possible to use the original cover medium for the detection process. Therefore, using statistical correlation is a better approach to detect the watermark signal in the cover medium for WSNs.

Table 2.10 Watermarking detection techniques used in literature

Author	Year	Watermark detection technique
Feng et al.	2003	$\varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4$
Sion et al.	2004	Similarity
Koushanfar et al.	2007	---
Albath et al.	2007	Calculate MAC over p(xi) and compare to MAC received with packet.
Zhang et al.	2008	A Gaussian of hypothesis testing on correlation coefficient
Juma et al.	2008	synchronization point or not
Xiao et al.	2008	Manipulation embedding bit and decrypt with key
Xiaomei et al.	2009	Mean and standard deviation
Xuejun	2010	---
Wang et al.	2011	By judging the parity of the LSB of each data.
Kamel et al.	2011	Comparison hash calculated between sender and receiver

2.6.9 Noise

Noise is the name given to any agent or incident which deteriorates the quality of communication between the sender and the receiver. Noise in WSN communication channels may be of many different kinds, e.g. failure of the node, probability of collusion, incoming data rate, and dropped packet data. Various types of noise used in the literature have been summarized in Table 2.11.

In the literature, most of the works do not mention the noise in their experiments (Kamel and Juma 2011), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011), (Jessica and Potkonjak 2003), (F. Koushanfar Auerbach publications 2007). In others, the noises mentioned are incoming quality data rate (Radu, Mikhail, and Sunil 2004), dropped packet data (Albath and Madria 2007), (Zhang, Liu, and Das 2008), and decrease in the quality of the transmitting data (Xiaomei and Xiaohua 2009). In our estimation, dropped packet data is a more dangerous type of noise than decrease in the quality of the

transmitting data or the incoming data rate, because it can cause the communication between the sender and the receiver to stop altogether.

Table 2.11 Different types of noise in literature

Author	Year	Noise
Feng et al.	2003	---
Sion et al.	2004	Incoming data rate
Koushanfar et al.	2007	---
Albath et al.	2007	---
Zhang et al.	2008	Node failure
Juma et al.	2008	--
Xiao et al.	2008	---
Xiaomei et al.	2009	Decrease in the quality of the transmitting data
Xuejun	2010	----
Wang et al.	2011	---
Kamel et al.	2011	---

2.6.10 Vulnerability Attacks

Vulnerability attacks are performed in watermark testing to estimate the strength of the watermarked cover medium, and then demodulate it by subtracting the estimated strength from some constant strength factor. The determination of strength factor is based on either of the two conditions – removal of maximum watermark energy, or minimization of the attacked cover medium's cross-correlation coefficient with the watermark signal. Following is a comparative evaluation of different vulnerability attacks used to test watermarking techniques in WSNs. All these have been summarized in Table 2.12.

Most of the works test their algorithms against 'man-in-the-middle attack', e.g. forgery, impersonation, modification of data, or insertion of false data (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Juma, Kamel, and Kaya 2008), (Xiaomei 2009), Wang et al. (Wang, Sun, and Ruan 2011). Ghost signature and de-synchronization (F. Koushanfar Auerbach publications 2007), and statistical attacks (Zhang, Liu, and Das 2008), (Xiaomei 2009), (Wang,

Sun, and Ruan 2011) are some other vulnerability attacks used. In the rest, no mention of the attacks used is found (Jessica and Potkonjak 2003), (Albath 2007), (Rong, Xingming, and Ying 2008), (Xuejun 2010). In our estimation, securing the algorithm against man-in-the-middle attacks is more important because these are the most often seen attacks on wireless sensor networks with the attackers trying to interfere with the communication channel in an attempt to exploit the vulnerabilities of the channel.

Table 2.12 Attack used in literature

Author	Year	Attacks
Feng et al.	2003	----
Sion et al.	2004	<ul style="list-style-type: none"> • Sampling • Segmentation • Linear changes • Addition stream
Koushanfar et al.	2007	<ul style="list-style-type: none"> • Ghost Signature • Addition new signature • Removal of the author's signature • De-synchronization
Albath et al	2007	---
Zhang et al	2008	<ul style="list-style-type: none"> • False distribution imposition all sensor • False distribution imposition part of sensor • Remnant check
Juma et al.	2008	<ul style="list-style-type: none"> • Modify data • Adding data false
Xiao et al.	2008	---
Xiaomei et al.	2009	<ul style="list-style-type: none"> • Personate • Forgery • Sampling • Summarization
Xuejun R	2010	---
Wang et al.	2011	<ul style="list-style-type: none"> • Forgery attack • Selective forward • Data replication • Packet transfer delay • Packet tempering
Kamel et al.	2011	---

This section gave a detailed survey and evaluation of various digital watermarking techniques for WSNs. The next section will do the likewise for WMSNs.

2.7 Survey of Literature on Digital Watermarking Technique in WMSNs

With the emergence of WMSNs, the focus in wireless communication networks has shifted from WSNs dealing with the typical scalar data to the networks that can gather multimedia data from the external environment. Most of the numerous applications of WMSNs are image-based. e.g. traffic monitoring and enforcement. Obviously, the multimedia sensor nodes in WMSNs have to capture, process and transmit huge amounts of data.

However, due to the technical specifications of WMSNs, these multimedia applications sometimes also create bottle-necks in the network traffic (Baowei Wang 2011). Moreover, as WMSNs share a common physical medium without any central management, and with limited computational resources, they are particularly vulnerable to security attacks.

Many approaches have been experimented with to ensure security in WMSNs, watermarking being one of them. In this technique, some data for identification, called watermark, is inserted in the cover medium, and later on extracted to identify and authenticate it (Wang 2010). Many researchers have proposed the concept and confirmed their watermarking techniques through simulation and/or experiments. However, different researchers have used different experiments with different settings in terms of sensed data, type of watermark used, type of domain and type of attack.

This section carries out a survey of the current literature on digital watermarking techniques for WMSNs, and evaluates the techniques proposed. To adequately analyse the existing literature, we evaluate all the algorithms on 10 parameters. These are, cover medium, sensed data, watermark signal, embedding key, transform domain, watermark generator, watermark embedding technique, watermark detection technique, noise, and attack.

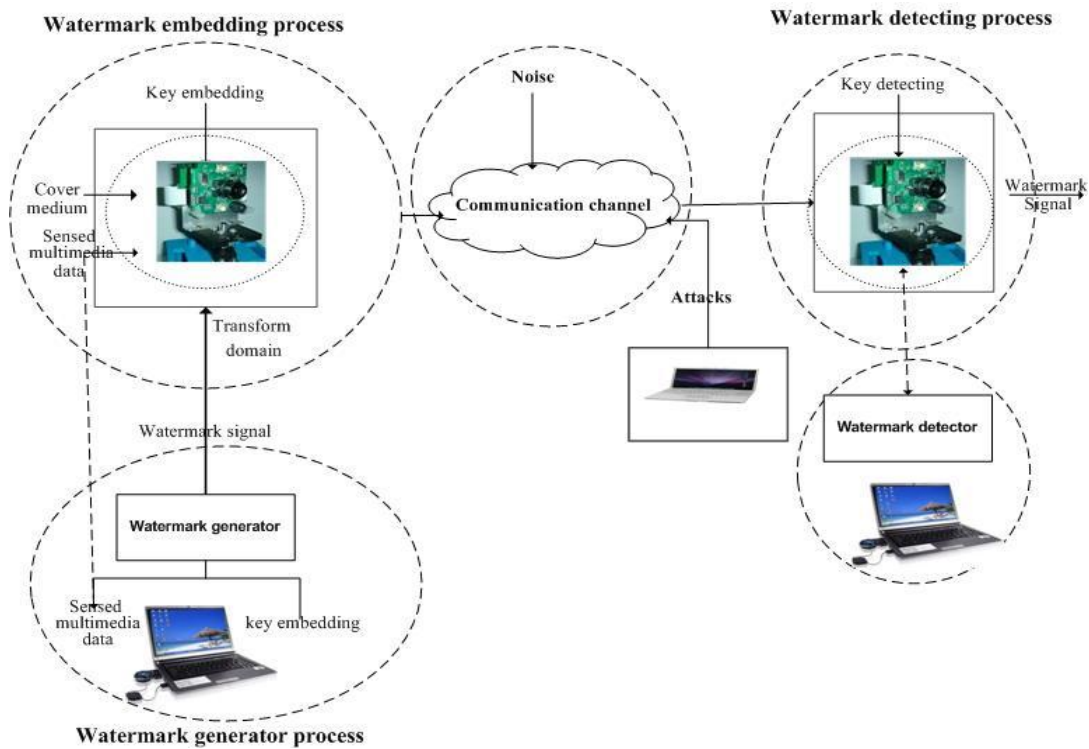


Figure 2.23 The 10 parameters for the evaluation of digital watermarking techniques in WMSNs

The 10 parameters for the evaluation of the digital watermarking techniques in WMSNs have been shown in Figure 2.23. The idea behind adopting this approach is to carry out an independent and thorough evaluation of each algorithm by clearly studying each aspect independently. We will study 6 different algorithms on these 10 parameters, beginning with cover medium.

2.7.1 Cover Medium

Cover medium is one of the key components of the watermarking technique, and is used for inserting a watermark signal. There are different types of cover media, such as texts, images, audio or video signals, and packet data, as presented in Figure 2.24.

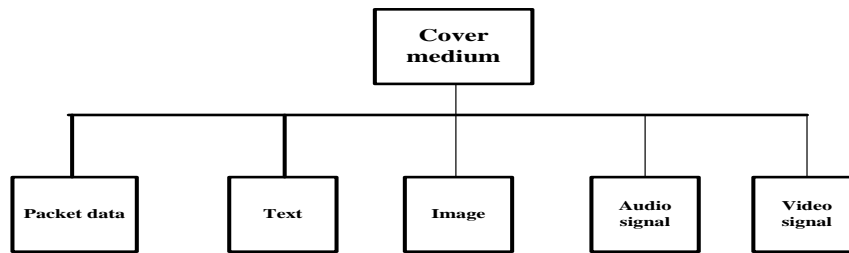


Figure 2.24 Cover media used in WMSN literature

Packet data and signals are used as cover media for watermarking techniques in WMSNs. Following is a comparative evaluation of different cover media. All of them have been shown in Table 2.13. Majority of these use ‘packet data’ as the cover medium (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang 2010), (Masood, Haider, and Sadiq ur 2010). However, some have used ‘audio signal’ (Padmavathi, Shanmugapriya, and Kalaivani 2010), (Kaur 2010). In this case, we cannot decide which one is better because it depends on the main objective of inserting the watermark signal. For example, Kaur (2010) uses a signal for inserting binary stream in which the binary stream is generated by an 8-bit chirp signal. This signal is used to protect the patient from the ECG signal. On the other hand, Honggang et al. (Honggang Wang, Dongming Peng, and Wei Wang 2008) and Wang et al. (Xiangjun, Shadong, and Le 2008) use packet data as the cover medium for embedding an image logo. The image logo is not produced by watermark generator, but it can be used to protect the cover medium. Both of them only consider the location of the image logo in the cover medium by using DWT.

Table 2.13 Cover media used in literature

Author	Year	Cover medium
Honggang et al.	2008	Packet data
Pingping et al.	2009	Packet data
Wang et al.	2010	Packet data
Padmavathi et al.	2010	Audio signal
Kaur et al.	2010	Audio signal
Masood et al.	2011	Packet data

2.7.2 Sensed Data

As mentioned in Section 2.5.3, the sensed data, like the watermark message, have to be communicated through an unsecure channel. However, a multimedia sensor node is more capable than a scalar sensor node, since it can retrieve not only scalar data but also video, audio, images and signals (Almalkawi, Guerrero Zapata, and Al-Karaki 2010). Here the sensed data generated by the multimedia sensor node need copyright protection as they are to be communicated to the other multimedia sensor nodes via an unsecure channel.




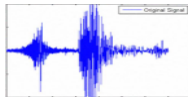
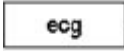

Honggang Wang, Dongming Peng, and Wei Wang (2008) and Wang (2010) implement the image data authentication with ‘the image’ as the sensed data. The optimized values in selective LH3/HL3 coefficients’ locations have been pursued by them to embed the watermark logo in the host multimedia. Pingping (2009) also implements copyright protection of the image with ‘the image’ as the sensed data. However, he has embedded the logo watermark in the low-frequency coefficients of DCT.

Masood, Haider, and Sadiq-ur (2010) implement security in WMSNs with ‘the image’ as the sensed data, encoded by a rate 5/8 encoder and then decoded by Viterbi decoder. Masood, Haider, and Sadiq-ur (2010) also use the watermarking technique to make communication between the sensor nodes secure.

Padmavathi, Shanmugapriya, and Kalaivani (2010) use ‘audio acoustic signal’ as the sensed data for identifying the category of the vehicle, by embedding the signal, while Kaur (2010) has embedded a patient’s identification signal for increasing the security of an ECG signal, with ‘the ECG signal’ as the sensed data.

Different kinds of sensed data have been compared over all the different approaches in Table 2.14. As can be seen, the majority of the approaches use ‘image’ as the sensed data (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang 2010), (Masood, Haider, and Sadiq ur 2010). However, some have also used ‘signal’ as the sensed data (Kaur 2010), (Padmavathi, Shanmugapriya, and Kalaivani 2010). In this case also, we cannot state which one is better, since it depends on whether the multimedia sensor node has captured image or signa.

Table 2.14 Sensed data used in literature

Author	Year	Sensed Data
Honggang et al.	2008	image 
Pingping et al.	2009	Image 
Wang et al.	2010	image 
Padmavathi et al.	2010	audio acoustic signal 
Kaur et al.	2010	ECG Signal 
Masood et al.	2011	Image 

2.7.3 Type of Watermarks

As mentioned in Section 2.5.4, a pattern of bits inserted in a cover medium is used as watermark. A watermark may consist of not only watermark constraints, watermark bits and binary streams, but also image logos, binary matrices and signals. Examples of these watermarks are given in Figure 2.25

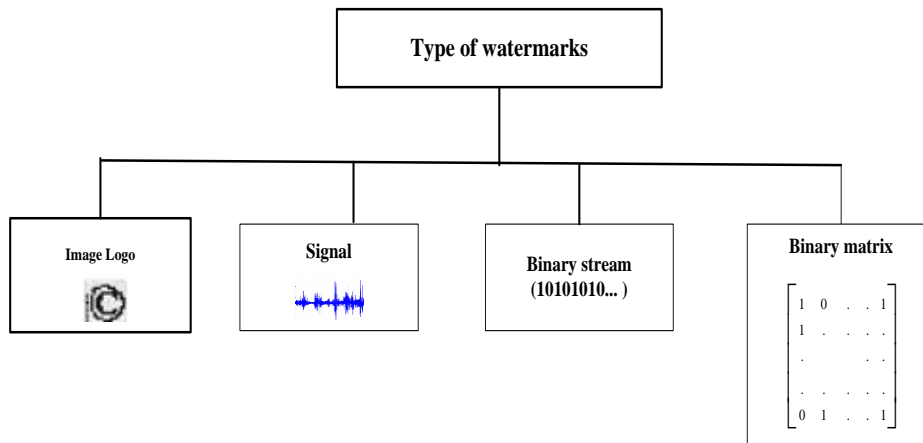




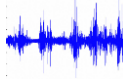


Figure 2.25 Types of watermarks for WMSNs

Following is a comparative evaluation of different types of watermarks in cover media meant for WMSNs. All the approaches have been summarized in Table 2.15.

The majority of these use ‘signal’ as a watermark (Masood, Haider, and Sadiq-ur 2010), (Padmavathi, Shanmugapriya, and Kalaivani 2010), while some use ‘image logo’ (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010). Only a few have used ‘binary matrix’ as watermark signal (Pingping, Yao Jiangtao, and Zhang Ye 2009), while one has used ‘binary stream’ for the purpose (Kaur 2010).

In our estimation, image logo is better than binary matrix as a watermark, because it can easily be detected and extracted. Statistical approaches, such as NC, Mean Squared Error (MSE) and Peak Signal Noise ratio (PSNR), can detect the image logo (Wenjun and Liu 1999), while IDWT, IFFT and IDCT can invert their domains. The image logo can also be separated from the cover medium and can be seen with naked eye. In addition, the image logo is a meaningful watermark because people can detect whether there is a watermark and identify it by visual observation.

Table 2.15 Types of watermark data used in literature

Author	Year	Type of watermarks
Honggang et al.	2008	Image logo 
Pingping et al.	2009	Binary matrix $\begin{bmatrix} 1 & 0 & \dots & 1 \\ 1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 1 \end{bmatrix}$
Wang et al.	2010	Image logo 
Padmavathi et al.	2010	signal 
Kaur et al.	2010	Signal 
Masood et al.	2011	Signal 

2.7.4 Watermark Key

Recalling Section 2.5.5, the embedding and detection processes use a key, called the watermark key, when the watermark signal is inserted into the cover medium. This key is also used to make the watermark secure, that is, prevent unauthorized parties from retrieving and changing the watermark. The watermark keys are the two adaptive threshold, the weight coefficient of the watermark, the user's insertion key and ID. Examples of the watermark keys have been given in Figure 2.26. The thresholds T_1 and T_2 have been selected at LH3/HL3 location. Filtering and deciding the appropriate embedding position has been done by two adaptive thresholds $\{T_1, T_2\}$ (Hani, Ernest, and Juan Gonzalez 2008). Thus, the embedding positions depend on these positions of the watermark. ID is $b_j = j^{th}$ bits of the patient ID in binary format where Kaur (2010) has used a patient ID. This ID has been used to modulate the chirp signal.

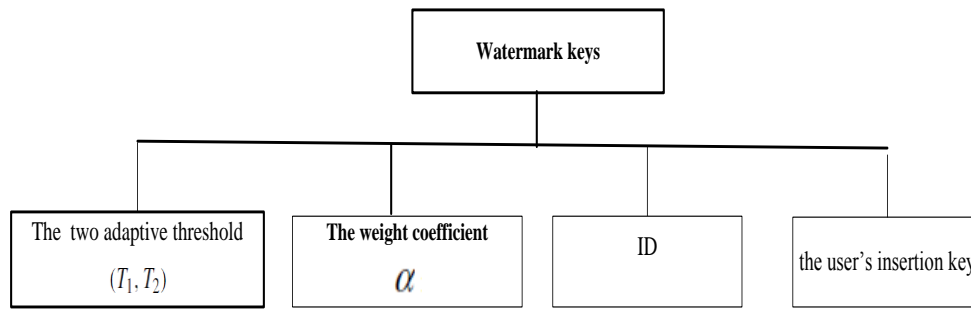


Figure 2.26 Types of watermark keys used in WMSNs

Following is a comparative evaluation of different types of watermark keys used in the literature. All of these have been summarized in Table 2.16. Two of the works surveyed use ‘the two adaptive threshold’ as the watermark key (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010). Others have used the coefficient of the watermark signal (Pingping, Yao Jiangtao, and Zhang Ye 2009), the user’s insertion key (Padmavathi, Shanmugapriya, and Kalaivani 2010) and patient ID (Kaur 2010), while one does not mention the watermark key used (Masood, Haider, and Sadiq-ur 2010). We believe that the two adaptive threshold $\{T_1, T_2\}$ is better than the coefficient of the watermark because it can filter and decide the appropriate embedding position (Honggang Wang, Dongming Peng, and Wei Wang 2008), while the coefficient of the watermark can’t be used for this purpose. On the other hand, the two adaptive threshold and the user ID cannot be compared because they have different purposes.

Table 2.16 The watermark keys used in literature

Author	Year	Watermark key
Honggang et al.	2008	Two adaptive threshold (Threshold key)
Pingping et al.	2009	Weight coefficient of the watermark signal
Wang et al.	2010	Two adaptive threshold (Threshold key)
Padmavathi et al.	2010	The user’s insertion key
Kaur et al.	2010	Patient ID binary digit (15 bit)
Masood et al.	2011	---

2.7.5 Transform Domain

The transform domain of digital watermarking techniques is normally classified into three types, i.e. Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT). Below, we provide a comparative evaluation of different types of transform domains used in the literature. All these have been summarized in Table 2.17. ‘DWT’ has been used as the transform domain by Honggang Wang, Dongming Peng, and Wei Wang (2008) and Wang (2010), ‘DCT’ by Pingping, Yao Jiangtao, and Zhang Ye (2009) and Kaur (2010), and ‘FFT’ by Masood, Haider, and Sadiq-ur (2010). Padmavathi, Shanmugapriya, and Kalaivani (2010), on the other hand, do not mention the transform domain used by them (Padmavathi, Shanmugapriya, and Kalaivani). In our estimation, DWT is more robust than DCT and FFT, because it can embed a watermark in the selective coefficients of the middle frequency bands of an image frame at the three-level Discrete Wavelet Transform (DWT), based on the network conditions.

Table 2.17 Transform domains used in literature


Author	Year	Domain
Honggang et al.	2008	DWT
Pingping et al.	2009	DCT
Wang et al.	2010	DWT
Padmavathi et al.	2010	--
Kaur et al.	2010	DCT
Masood et al.	2011	FFT

2.7.6 Watermark Generator

As elaborated in Section 2.5.6, creating any kind of watermark requires a watermark generator. Here, we evaluate different watermark generators used in the literature over all our parameters. All the approaches have been summarized in Table 2.18. A watermark generator for watermarking in WMSNs consists of a median filter, an 8-bit chirp signal, and a 5/8 encoder block. The median filter relies on data, instead of means, the 8-bit chirp signal is a ‘quadratic function’ with the

frequency sweep $f_i(t) = f_o + \beta t^2$, and the 5/8 encoder block is a non-recursive encoder, in which the input is encoded and included in the output sequence. The majority of the works surveyed do not mention the generator used (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang 2010). Others have used median filter (Padmavathi, Shanmugapriya, and Kalaivani 2010), the 8-bit chirp signal (Kaur 2010) and the 5/8 encoder block (Masood, Haider, and Sadiq ur 2010). However, it is not possible to meaningfully compare these watermark generators as all are used for different purposes. The median filter generator produces a watermark signal with the signal and user key insertion as input. On the other hand, a watermark binary stream (Masood, Haider, and Sadiq ur 2010) is generated by the 8-bit chirp form with signal and patient ID as input. The 5/8 encoder block generator produces a watermark signal from image. However, this generator only uses images as input and there is no mention of the watermark key used either.

Table 2.18 Watermark generators used in literature

Author	Year	Watermark generator
Honggang et al.	2008	---
Pingping	2009	---
Wang et al.	2010	---
Padmavathi et al.	2010	Median filter is used to denoise the signal
Kaur et al.	2010	The 8-bit chirp signal 
Masood et al.	2011	(5/8 Encoder Block)

2.7.7 Watermark Embedding Technique

As discussed in Section 2.5.7, there are four techniques for watermark embedding in WMSNs, i.e., using an LSB, using an MSB, using the DSSS system and adding watermark constraints. However, watermarking techniques for WMSNs differ from those for WSNs. Below, we provide a comparative evaluation of different kinds of watermarking techniques for WMSNs used in the literature. All these techniques have been summarized in Table 2.19.

In two of the works, ‘the two filter adaptive threshold’ has been used as an inserting technique (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010) in DWT. Others have

used varied techniques, viz. the weight coefficient of the watermark in DCT (Pingping, Yao Jiangtao, and Zhang Ye 2009), Orthogonal Frequency Davison Multiplexing (OFDM) in FFT (Masood, Haider, and Sadiq-ur 2010) and the Wiener Filter (Padmavathi, Shanmugapriya, and Kalaivani 2010). Figure 2.27 (a) shows the two filter adaptive threshold, Figure 2.27 (b) shows the DCT with the weight coefficient of the watermark information and location, and Figure 2.27 (c) shows the OFDM

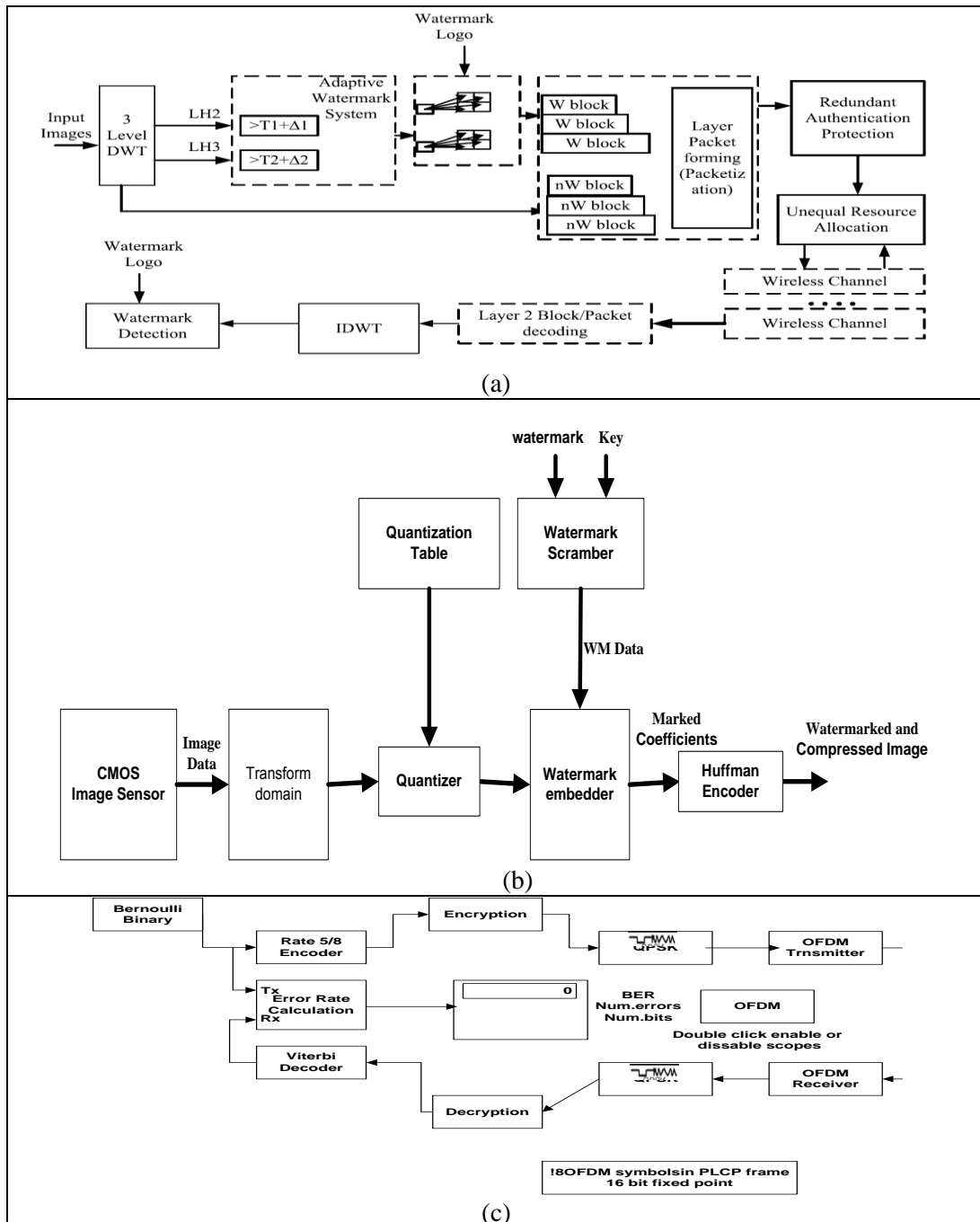


Figure 2.27 (a) The two adaptive threshold (Honggang et al. 2008), (b) the weight coefficient of the watermark in DCT (Pingping, Yao Jiangtao, and Zhang Ye 2009) and (c) Orthogonal Frequency Davison Multiplexing (OFDM) in FFT (Masood, Haider, and Sadiq-ur 2010)

The desired or targeted random process is estimated using the Wiener Filter, by filtering another random process through it. The input for the Wiener filter is taken to be a signal $s(t)$, corrupted by an additive noise $n(t)$. The output $\hat{s}(t)$ is calculated by means of a filter $g(t)$, using the following convolution:

$$\hat{s}(t) = g(t) * [s(t) + n(t)] \quad (2.17)$$

We find ‘the two filter adaptive threshold’ better as an inserting technique than others, like the weight coefficient of the watermark and OFDM, because the two filter adaptive threshold technique uses the three level DWT and is, therefore, energy efficient, as the adaptive threshold positions to insert watermark are dynamically chosen according to network conditions (Wang 2010).

Table 2.19 Watermarking embedding technique used in literature

Author	Year	Watermark embedding technique
Honggang et al.	2008	F_w is the watermarking function and $\{p_1, p_2, \dots, p_{mxw}\}$ is a set of positions where the watermark is embedded under the specific watermarking schema. The two adaptive threshold $\{T_1, T_2\}$ is used to filter the appropriate embedding process. PLR is the packet loss ratio. So, the function is $D = F_{watermark}(\{p_1(T_1, T_2), \dots, p_i(T_1, T_2)\}, PLR)$
Pingping et al.	2009	The embedding algorithm has the watermark equation $DCT'(p, q) = sign(DCT(p, q) + (\alpha * w))$ where w is the watermark data, α is the weight coefficient of the watermark information and (p, q) is the location.
Wang et al.	2010	The function $\{Q_w, Q_d\} = F_w(\{p_1(T_1, T_2), p_2(T_1, T_2), \dots, p_i(T_1, T_2)\}, PLR)$
Padmavathi et al.	2010	The process of embedding digital data in the form of $X' = Ek(X'W)$, where X is the pre-processed original

		signal, W is the watermark information being embedded, k is the user's insertion key
Kaur et al.	2010	The function is $f_i(t) = f_o + \beta t^2$ where $\beta = (f_1 - f_0)t^{-2}$. $y_{chirp, mod} = y_{chirp} * f(b_j)1$ b_j is patient ID y is the watermarked signal
Masood et al.	2011	The embedding process is $d_w = E(d_o, k, w)$. d_o original w watermark message k security key

2.7.8 Watermark Detection Technique

As mentioned in Section 2.5.8, the detection process consists of an extraction unit to first extract the watermark signal and later compare it with the cover medium. The process of extraction or detection is used for checking whether there is a watermark signal in the cover medium or not. Below, we provide a comparative evaluation of this process used for watermarking in WMSNs, as found in the literature. All the different approaches have been summarized in Table 2.20.

As can be seen from the table, most of the works use ‘statistical approach’ for detection, e.g. Normalized Correlation (NC) and Peak Signal-to Noise Ratio (PSNR) (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Padmavathi, Shanmugapriya, and Kalaivani 2010). However, ‘the 8-bit chirp signal’ has also been used. In our estimation, the statistical approach scores over the 8-bit chirp signal because it is a more common and valid method used for detection of the watermark, without medium signal. Using the medium signal for detection is not possible in WMSNs, because the watermark image is invisible to the naked eye.

Table 2.20 Watermark detecting technique used in literature

Author	Year	Watermark detection technique
Honggang et al.	2008	To detect the watermark image, the normalized correlation (NC) coefficient is used to measure similarity of original watermarks and extracted watermark
Pingping et al.	2009	Peak signal-to Noise ratio (PSNR) and the extracted watermark is obtained by comparing d with 0, i.e., $w' = \begin{cases} w & d=0 \\ 0 & d>0 \end{cases}$.
Wang et al.	2010	Normalized correlation (NC) $NC = \frac{\sum_{i=1}^w \sum_{j=1}^m w(i, j) w^*(i, j)}{\sum_{i=1}^w \sum_{j=1}^m [w(i, j)]^2}$
Padmavathi et al.	2010	Mean Square Error (MSE) and Peak Signal-to Noise ratio (PSNR)
Kaur et al.	2010	---
Masood et al.	2011	The watermark detection process is defined as $w_e = D(d_w, k, w, d_o)$ watermarked data original data and watermark w ; Key k ;

2.7.9 Noise

As mentioned in Section 2.5.8, noise can be defined as anything that interferes with the communication channel, deteriorating the quality of communication. Kinds of noise include packet loss, decrease in the quality of transmission, and packet drop. Below, we provide a comparative evaluation of different types of noise for watermarking in WMSNs as found in the literature. All of these are summarized in Table 2.21.

Two of the researchers use ‘dropped packet data’ as the noise in their experiments (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010), while one has used ‘pepper salt’ (Pingping, Yao Jiangtao, and Zhang Ye 2009). Others do not mention the type of noise used by them (Masood, Haider, and Sadiq ur 2010). We believe that the dropped packet loss is a more dangerous type of noise because it can cause the communication between the sender and the receiver to stop altogether. To overcome this noise, the sender has to retransmit the packet data which, however, wastes energy.

Table 2.21 Different types of noise used in literature

Author	Year	Noise
Honggang et al.	2008	Packet loss
Pingping et al.	2009	Paper salt noise
Wang et al.	2010	Packet loss
Padmavathi et al.	2010	---
Kaur et al.	2010	Any undesirable noise
Masood et al.	2011	Noise communication

2.7.10 Vulnerability Attacks

The main objective of vulnerability attacks is to modify, delete or remove any watermark signal from the cover medium. There are two types of attacks: intentional and accidental. Intentional attacks include cryptanalysis, image processing technique, and removing the existing watermark. On the other hand, some undesirable consequences of standard image processing, e.g. resizing, filtering or compressing, are parts of accidental attacks. The different attacks used for testing the watermarking techniques in WMSNs belong to the category of accidental attacks, such as cropping, filtering and compression. Below, we provide a comparative evaluation of different vulnerability attacks used by different researchers. All these have been summarized in Table 2.22. In the works of most of these, ‘accidental’ attacks, such as cropping and compressing, have been used (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010), (Pingping, Yao Jiangtao, and Zhang Ye 2009), while in one of these ‘filtering’ has been used (Kaur 2010). Others do not mention the kind of attack used for their watermarking techniques (Padmavathi, Shanmugapriya, and Kalaivani 2010). We believe that cropping and compressing have greater possibility to occur than filtering, since they belong to the category of accidental attacks.

Table 2.22 Different types of attack used in literature

Author	Year	Attack
Honggang et al.	2008	Cropping and compression
Pingping et al.	2009	Cropping and compression
Wang et al.	2010	Compression
Padmavathi et al.	2010	---
Kaur et al.	2010	Filtering
Masood et al.	2011	---

This concludes the survey and evaluation of digital watermarking technique for WMSNs found in the literature. The next section provides a summary of approaches and issues in digital watermarking technique for both WSNs and WMSNs.

2.8 Summary of Issues in Digital Watermarking for WSNs and WMSNs

As previously stated, the main aim of this study is to develop a digital watermarking technique for both WSNs and WMSNs. To be able to do this, we evaluated the literature on digital watermarking techniques for both WSNs and WMSN. This comparative evaluation on five security parameters has been shown in Table 2.23.

Table 2.23 Comparison of different issues in secure communication between sensor nodes for WSNs

Author	Year	Secure Communication	Data Authentication	Integrity	Copyright Protection	Privacy
Feng et al.	2003	X	X	X	√	X
Sion et al.	2004	x	√	x	x	x
Koushanfar et al.	2007	x	x	x	√	x
Albath et al	2007	√	x	x	x	x
Zhang et al	2008	x	√	x	x	x
Juma et al.	2008	x	x	√	x	x
Xiao et al.	2008	x	x	x	√	x
Xiaomei et al.	2009	x	√	x	x	x

Xuejun R	2010	X	√	X	X	X
Wang et al.	2011	√	X	X	X	√
Kamel et al.	2011	X	X	√	X	X

√ Yes X No

As can be seen from Table 2.25, Xiangqian (2009) and Wang, Sun, and Ruan (2011) present issues in secure communication between sensor nodes in WSNs, while Radu, Mikhail, and Sunil (2004), Zhang, Liu, and Das (2008), (iaomei and Xiaohua (2009), and Xuejun (2010) provide issues of data authentication in WSNs. Xuejun (2010) and Kamel et al. (2011), on the other hand, present issues of data integrity in WSNs. The issue of copyright protection has been taken up by Jessica and Potkonjak (2003), Koushanfar and Potkonjak (2007) and Rong, Xingming, and Ying (2008), while the issue of privacy has been discussed by Wang et al. (2011).

Thus, we made a comparative evaluation of the different aspects of digital watermarking technique for WSNs along 11 different algorithms on 10 different parameters. This evaluation can be summarized across each parameter as follows:

1. Cover medium: Most of the researchers have used 'packet data' as the cover medium (Kamel 2011), (Radu, Mikhail, and Sunil 2004), (Xiangqian 2009), (Zhang, Liu, and Das 2008), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xiaomei and Xiaohua 2009), (Xuejun 2010), (Wang, Sun, and Ruan 2011).
2. Watermark message: Most often text messages have been converted into binary streams using 'text' as the watermark message (Jessica and Potkonjak 2003), (Koushanfar and Potkonjak 2007). We find 'text', such as 'do not mention anything' to be a better watermark message.
3. Sensed data: Most of the researchers have used 'sensed data' of binary stream with 8 bits and 16 bits as the message (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011).
4. Type of watermark signal: Most of the works use 'binary stream' as the watermark signal (8, 16 , 64 , 128 bits) (Kamel and Juma 2011), (Radu, Mikhail, and Sunil 2004), (Albath 2007),

-
- (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011).
5. Watermark key: Most of the works use 'binary stream' for watermark key (Kamel and Juma 2011), (Albath and Madria 2007), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009). Some of them suggest using positive integer (Radu, Mikhail, and Sunil 2004), and amplitude (Zhang, Liu, and Das 2008).
 6. Watermark generator: Most of the reseachers use 'hash function' for watermark generation, such as MD5 and SHA (Kamel and Juma 2011), (Fang Jessica 2003), (F. Koushanfar 2007), (Albath 2007), (Juma, Kamel, and Kaya 2008), (Xiaomei and Xiaohua 2009).
 7. Watermark embedding technique: 'Least Significant Bits' (LSB) has been used as the most frequent embedding technique (Radu, Mikhail, and Sunil 2004), (Albath and Madria 2007), (Zhang, Liu, and Das 2008), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011), (Kamel and Juma 2011).
 8. Watermark detection technique: 'Statistical correlation', e.g. probability, similarity, mean deviation, standard deviation, and Gaussian hypothesis, have been used by most for detection (Radu, Mikhail, and Sunil 2004), (Albath and Madria 2007), (Zhang, Liu, and Das 2008), (Rong, Xingming, and Ying 2008), (Xiaomei 2009), (Xuejun 2010).
 9. Noise: Most researchers make no mention of the noise used in their experiments (Kamel and Juma 2011), (Juma, Kamel, and Kaya 2008), (Rong, Xingming, and Ying 2008), (Xuejun 2010), Wang et al. (Wang, Sun, and Ruan 2011), (Fang Jessica 2003), (F. Koushanfar 2007). However, we consider the dropped packet data as the most dangerous.
 10. Vulnerability attacks: 'Man-in-the-middle attack', e.g. impersonation, forgery, modification of data or insertion of false data, are the vulnerability attacks most frequently used (Juma, Kamel, and Kaya 2008), (Radu, Mikhail, and Sunil 2004), (Kamel and Juma 2011), (Xiaomei and Xiaohua 2009), (Wang, Sun, and Ruan 2011). Ghost signature, de-synchronization (Koushanfar and Potkonjak 2007), and statistical attack (Zhang, Liu, and Das 2008), (Xiaomei and Xiaohua 2009) (Wang, Sun, and Ruan 2011) have also been used
-

We also made a comparative evaluation of different aspects of digital watermarking technique in WMSNs on different parameters. This comparative evaluation has been summarized in Table 2.23 pertaining to the issues in secure communication between multimedia sensor nodes in WMSNs.

As can be seen in Table 2.26, Kaur (2010) and Masood, Haider, and Sadiq-ur (2010) present issues in secure communication between multimedia sensor nodes in WMSNs. While Honggang Wang, Dongming Peng, and Wei Wang (2008) and Wang et al. (2010) discuss the issue of image authentication, Padmavathi, Shanmugapriya, and Kalaivani (2010) raise the issue of signal authentication. The issue of copyright protection has been taken up by Pingping, Yao Jiangtao, and Zhang Ye (2009), and the issue of privacy only by Kaur (2010).

Table 2.26 Comparison of different issues in secure communication between multimedia sensor nodes for WMSNs

Author	Year	Secure Communication	Authentication		Copyright Protection	Privacy
			Image	Signal		
Honggang, et.al	2008	X	√	X	X	X
Pingping et.al .	2009	X	X	X	√	X
Wang et.al	2010	X	√	X	X	X
Padmavathi, et al	2010	X	X	√	X	X
Kaur, S et al	2010	√	X	X	X	√
Masood, et al	2011	√	X	X	X	X

√ Yes X No

We surveyed 6 works in the literature on watermarking technique for WMSNs across 10 different parameters. A summary of this evaluation on each parameter is as follows:

1. Cover medium: Most researchers have used 'packet data' as the cover medium (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang et al. 2010), (Masood, Haider, and Sadiq-ur 2010).

-
2. Sensed data: Most have used 'image' of binary stream with 8 bits and 16 bits as the sensed data (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang et al. 2010), (Masood, Haider, and Sadiq-ur 2010).
 3. Type of watermark: The majority of the researchers have used 'signal' as the watermark (Masood, Haider, and Sadiq-ur 2010), (Padmavathi, Shanmugapriya, and Kalaivani 2010), (Kaur 2010).
 4. Watermark key: 'The two adaptive threshold' has most commonly been used as the watermark key (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang et al. 2010). However, some researchers suggest using a weight coefficient of the watermark (Pingping, Yao Jiangtao, and Zhang Ye 2009), the user's insertion key (Padmavathi, Shanmugapriya, and Kalaivani 2010), and patient ID (Kaur 2010).
 5. Transform domain: As we saw, 'DWT' is the most commonly used transform domain (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang et al. 2010).
 6. Watermark generator: Unfortunately, most researchers do not mention the generator used by them (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Wang et al. 2010).
 7. Watermark embedding technique: In two of the works, 'the two filter adaptive threshold' has been used as an inserting technique (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang et al. 2010) in DWT.
 8. Watermark detection technique: Most works have utilized the 'statistical approach', such as Normalized Correlation (NC) and Peak Signal-to-Noise Ratio (PSNR), for watermark detection (Honggang Wang, Dongming Peng, and Wei Wang (2008), (Wang et al. 2010), (Pingping, Yao Jiangtao, and Zhang Ye 2009), (Padmavathi, Shanmugapriya, and Kalaivani 2010).
 9. Noise: The majority uses 'dropped packet data' as the noise for their experiments (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010), (Padmavathi, Shanmugapriya, and Kalaivani 2010).
-

10. Vulnerability attacks: Most of the works have utilized ‘accidental’ vulnerable attacks, such as cropping and compressing (Honggang Wang, Dongming Peng, and Wei Wang 2008), (Wang 2010), (Pingping, Yao Jiangtao, and Zhang Ye 2009), while one of them suggests using ‘filtering’ (Kaur 2010).

2.9 Conclusion

This chapter presented the mathematical background to be used in chapters 5, 6 and 7. It also presented a detailed survey of the current state-of-the-art digital watermarking techniques for WSNs and WMSNs. It further covered the issues in secure communication between sensor nodes for WSNs, and multimedia sensor nodes for WMSNs. It outlined the major issues with attempted solutions and paved the way for further investigation. Chapter 3 will now provide the problem definition and the research issues to be addressed in the rest of the thesis.

CHAPTER THREE

PROBLEM DEFINITION

This chapter covers:

- ▶ problem definition for copyright protection of scalar data in WSNs,
- ▶ problem definition for copyright protection of images in WMSNs,
- ▶ research issues to be taken into account,
- ▶ research methodology to be adopted in order to systematically address the research issues identified.

3.1 Introduction

The first chapter of the paper highlighted the major security issues in the copyright protection of scalar data as well as images. On the other hand, the second chapter provided an introduction to the mathematical background, the security services in WSNs and WMSNs and preliminary digital watermarking techniques, as well as a comprehensive survey and detailed evaluation of the latest digital watermarking techniques in both WSNs and WMSNs. It identified the weaknesses inherent in the current approaches addressing these issues. It also brought out the fact that, although there have been some significant contributions aiming to address these issues, no approach resolves the issues to any significant extent. While no single approach completely resolves the issue of copyright protection of scalar data in WSNs, very little work has been done for copyright protection of multimedia data (such as images) in WMSNs.

This chapter clearly delineates the problems which this research sets out to resolve, i.e., copyright protection of scalar data in WSNs and of images in WMSNs.

3.2 Problem definition

This section outlines the main problems and challenges in digital watermarking approaches as illustrated in Figure 3.1 and Figure 3.2. While Figure 3.1 shows the approach for copyright protection of scalar data between the sensor nodes of WSNs, Figure 3.2 shows the approach for copyright protection of images between the sensor nodes of WMSNs.

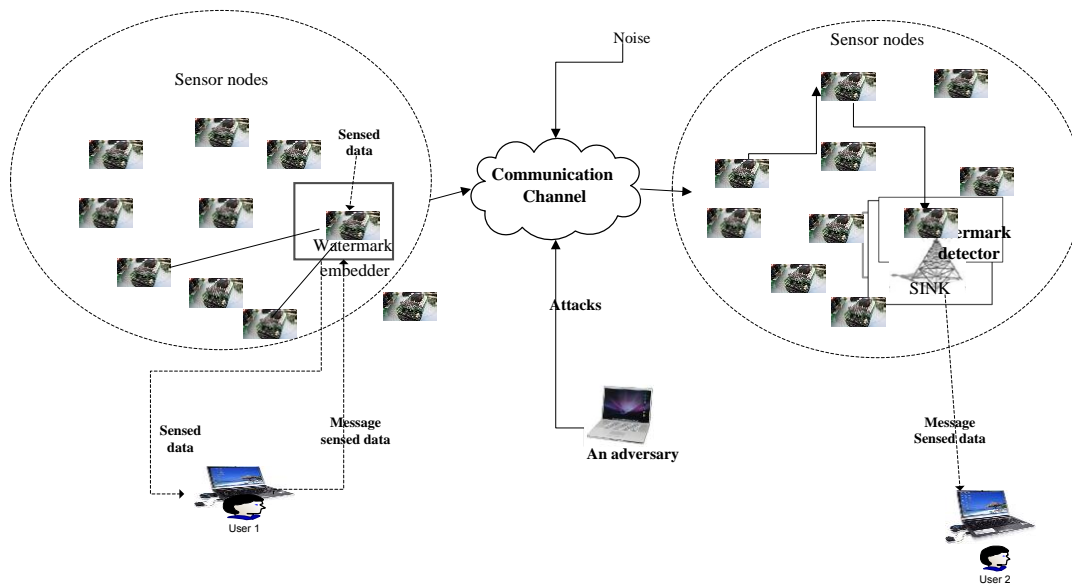


Figure 3.1 Copyright protection of scalar data between sensor nodes in WSNs

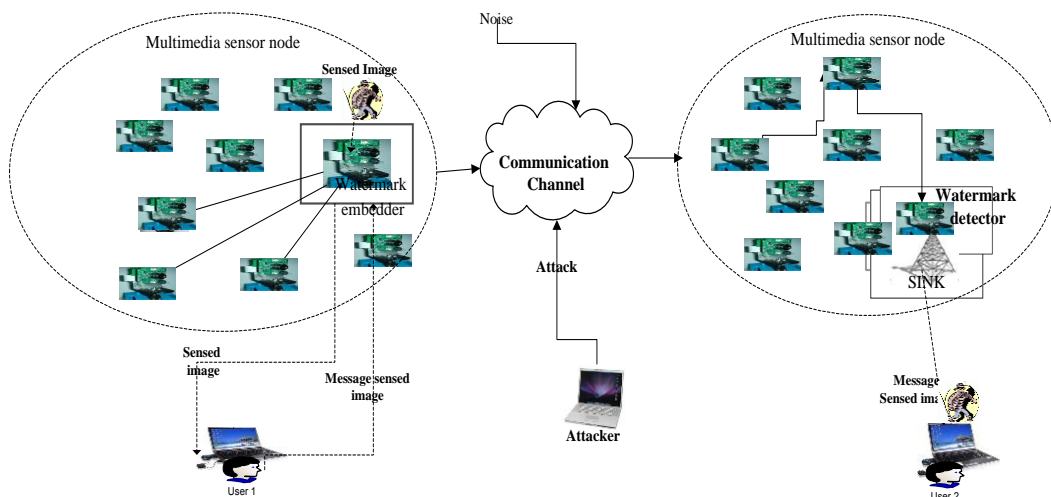


Figure 3.2 Copyright protection of images between multimedia sensor nodes in WMSNs

The problems and challenges associated with the security issues will be dealt with under the following two heads:

- Problems with copyright protection of scalar data in WSNs
- Problems with copyright protection of images in WMSNs

Both of these problems will be dealt with in the light of the economic and social concerns, the existing solutions, and the technical problems that these solutions involve, because these will be the key requirements for any effective and viable solution.

3.2.1 Problems with copyright protection of scalar data in WSNs

Economic and social concern

A Wireless Sensor Network (WSN) collects and stores the data sent to other nodes or servers. For example, suppose an electricity company wants to prevent electricity from being stolen. Now, since electricity is measured as scalar data, the company needs to protect it by adding a watermark signal to this scalar data. So, the company uses a WSN which is equipped with a watermark signal. The WSN is used to capture and store the scalar data in the areas suspected of rampant electricity theft. The data is then sent to the state electricity company data server through wireless networks installed in hostile, unattended environments. However, since WSNs are vulnerable to several different types of attacks, such as data insertion, data modification, and data repetition, security becomes an important issue with them. The traditional algorithms are not suitable for WSNs and cannot protect the copyright of the valuable scalar sensor data. The watermark acts as a second line of defence to ensure that the data is still valid. Therefore, the electricity company would hire a commercial entity with the aim of getting the original data from the consumers. The commercial interests of this entity will prompt it to ensure that it maintains the original data and is able to prove its authenticity.

Existing solutions

Several schemes have been proposed with a view to ensuring that the scalar data received is exactly as it was sent by the originator, with no tampering during the transit through the communication channels (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002, Ahmed et

al. 2002, Agah, Basu, and Das 2006) (Jian and Xiangjian 2005) (Ajay Jangra 2010) (Jessica and Potkonjak 2003) (Koushanfar and Potkonjak 2007) (Rong, Xingming, and Ying 2008) (Pingping, Yao Jiangtao, and Zhang Ye 2009). The majority of them use a digital watermarking technique that works through the Internet, and the peer-to-peer infrastructure that uses an unlimited amount of energy (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002, Ahmed et al. 2002, Agah, Basu, and Das 2006) (Jian and Xiangjian 2005) (Ajay Jangra 2010), although some of them work on WSNs that use limited power and capability (Jessica and Potkonjak 2003) (Koushanfar and Potkonjak 2007) (Rong, Xingming, and Ying 2008). One of the existing copyright protection models of scalar data, proposed by Rong, Xingming, and Ying (2008), has been described below to explain how a generic watermarking scheme works for WSN scalar data. Consider the copyright protection of scalar data scheme as shown in Figure 3.3.

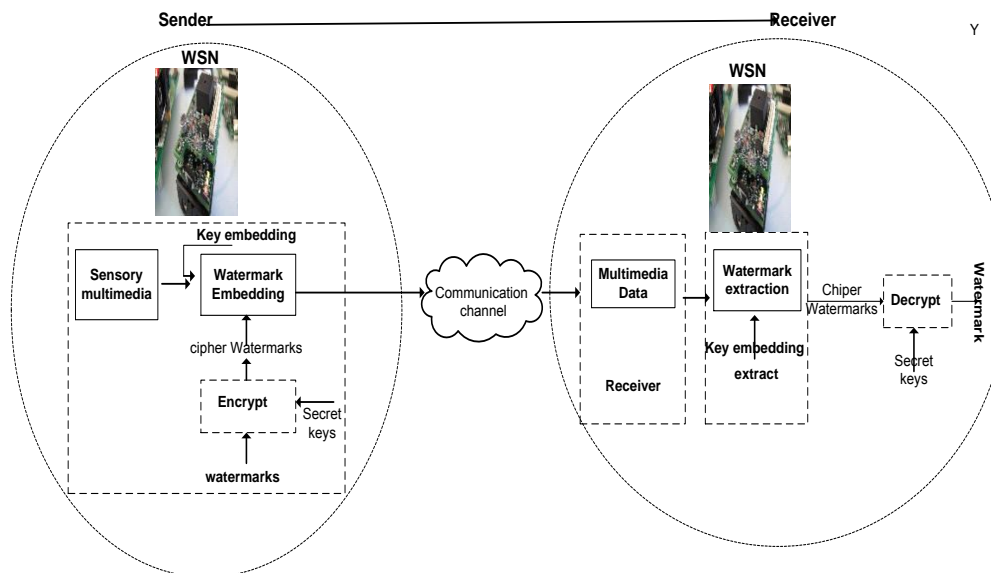


Figure 3.3 Copyright protection of scalar data in WSNs

This scheme needs several packets of sensory scalar data before embedding watermarks into these packets. These watermarks are encrypted by using secret keys to get cipher watermarks before they are embedded into these packets. They are then placed in a covert embedding position in sensing the scalar data packet. After this, all the packets that have the chipper watermark inserted are transmitted through a communication channel. The received scalar data is extracted using the embedded key to get the cipher watermark. Finally, the cipher watermarks are decrypted by using the secret key to get the watermarks.

Technical Problems

However, the scalar data in a WSN are transmitted from unattended and sometimes even hostile environments. So, it needs to be ensured for copyright protection that the data received through the sensor nodes have not been manipulated or modified during transit through the communication channel. The security systems traditionally used are computationally exhausting because of the overheads they impose, shortening the lives of the sensor nodes (Adrian, Szewczyk J. D. Tygar Victor, and Wen David 2002, Ahmed et al. 2002, Agah, Basu, and Das 2006). In comparison, the digital watermarking techniques are computationally light, neither imposing too many overheads nor requiring too much energy. However, the main drawback with this approach, as illustrated in Figure 3.3, is that the watermarks are encrypted before they are embedded into the packets of scalar data in WSNs. These encrypted watermarks need to execute thousands or even millions of instructions of multiplication to become the chipper watermark that can be embedded into the packets and sent through the wireless networks in hostile or unattended environments. Apart from this, the approach suffers from these shortcomings.

- **Inability to overcome the problem of duplicate sensory scalar data:** The copyright of the sensory scalar data needs protection against adversaries who can easily duplicate the sensor scalar data through the communication channel.
- **Inability to overcome the severe constraints on the sensory scalar data:** The mechanism of the copyright protection of scalar data, as shown in Figure 3.3, does not take into account the severe constraints on security, such as different types of attacks during transmission through the communication channel, e.g., replication, modification, and Sybil attacks. Therefore, the mechanism needs to be developed further against these attacks.

3.2.2 Problem with copyright protection of images in WMSNs

Economic and Social Concerns

Surveillance and remote monitoring is one of the applications of WMSNs. The system is managed by a commercial entity, with commercial interests, that captures the images of the people involved in unlawful activities in the business area using a WMSN. However, since the images are transmitted from unattended and sometimes even hostile environments, there is a high likelihood of these people, or others with malicious intent, accessing and modifying, manipulating, or deleting

these images from the WMSN. Therefore, the commercial entities managing such systems are paying greater attention to the issue of copyright protection of images in WMSNs. They embed watermarks to the images to ensure the authenticity of these images as well as prove their ownership in a court of law.

Existing Solutions

The current technology allows validation during transit but not after the image has reached its destination. Therefore, one of the challenges with these technologies is to ensure that the source of image is preserved after it has left the WMSN. Several schemes have been worked on for authentication of images in WMSNs (Honggang Wang, Dongming Peng, and Wei Wang 2008) (Xiangjun, Shaodong, and Le 2008) and authentication of signals (Padmavathi, Shanmugapriya, and Kalaivani) (Kaur 2010) (Masood, Haider, and Sadiq ur 2010) to ensure secure communication in WMSNs. However, there is no significant work as yet on copyright protection of images in WMSNs. Following is the existing model for copyright protection of images proposed by Pingping, Yao Jiangtao, and Zhang Ye (2009) in WSNs (Figure 3.4):

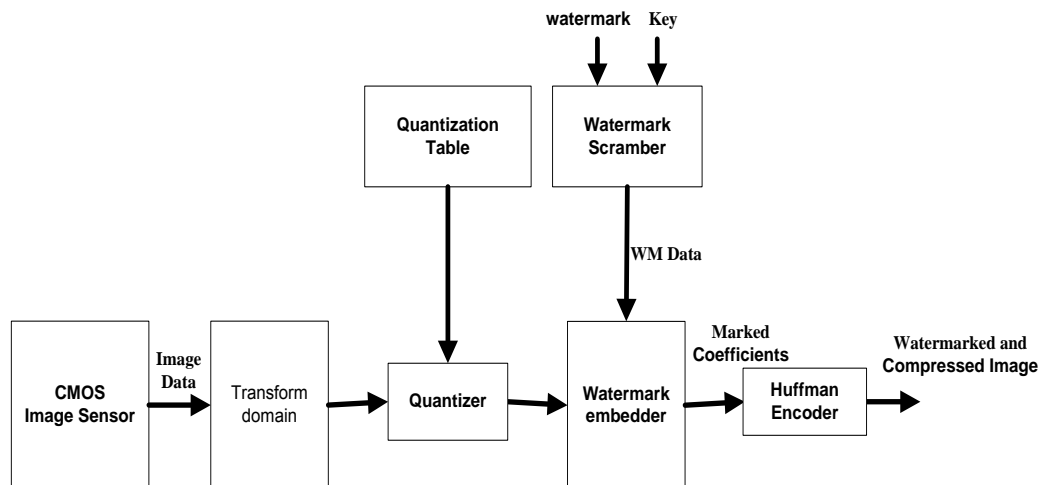


Figure 3.4 The architecture of CMOS image sensor system performing in the transform.

We now explain how a generic watermarking scheme works for copyright protection of images in WSNs (Pingping 2009). Consider the CMOS image sensor system architecture performing in the transform domain, as shown in Figure 3.4. Besides CMOS imager and transform domain, it consists of a Quantizer, watermark embedder, watermark scrambler and Huffman encoder. The CMOS imager produces an imager sensory array giving out a large number of pixels. The images

are formed by the sequences of digital values of the raw pixel data coming from these pixels. The sequences of digital values are transformed into the transform domain in order to get the coefficient transform domain. The coefficient transform domain is then quantized according to the quantization table in order to get the quantizer. The watermark being a binary matrix, the watermark data is first scrambled by the watermark scrambler using an initial secret key. Next, the watermark data is embedded by the watermark embedder using the quantizer. The watermark of the coefficient transform domain of the image is encoded using Huffman encoder in order to get a compressed watermarked image. The process of the extraction of watermark is the reverse of the process of embedding it. First, the detector decodes the watermarked and compressed image to get the watermarked image. The watermarked image is then extracted by comparing the coefficient transform domain with 1. Now, the low frequency coefficient in the transform domain is modified and the trade-off between imperceptibility and robustness is determined to obtain the location and , the weight coefficient of the watermark, in order to test the algorithm. As a result, the algorithm develops a stronger robustness against various attacks, e.g., compression, cropping, or noise insertion.

Technical Problem

Since the images are sent from unattended and sometimes even hostile environments, there is a high risk of intruders accessing the image sensors through wireless networks. Pingping (2009) embedded the watermark in the low-frequency coefficient of the transform domain. However, the main drawback with this approach, illustrated in Figure 3.4, is that it is not clear where the process of embedding or extracting the watermark is undertaken.

In summary, the key problems are:

- **Lack of clarity:** It does not make it clear as to how and where the process of embedding and extracting are to be undertaken during the image transmission between the sensor nodes.
 - **Inability to overcome the problem of duplicate images:** The copyright protection of the image has to be developed in a way to protect against adversaries who can easily duplicate segments of the images.
-

- **Inability to overcome the constraints on the copyright protection of the images:** The copyright protection of image model has been developed using a system of CMOS image sensors. This mechanism does not take into account the severe security constraints, such as different kinds of attacks during image transmission in the communication channel, e.g. replication, modification, and manipulation.

3.3 Research Issues

We have so far discussed two areas of the problem definition as explained above. For each of these areas, we defined the economic and social concerns, the currently available solutions, and the unresolved technical problems with these solutions. The technical problems form the basis of the research issues, and it is crucial to determine how they can be addressed and what the requirements for their solutions are. This section outlines the issues that the development of any new solution needs to take into account, and then, from Chapter 4 onwards, the solutions that attempt to address these research issues and objectives have been pursued.

3.3.1 Research Issue 1: Developing Copyright Protection of Scalar data in WSNs using Watermarking Technique

Chapter 2 (Section 2.5) presented a survey of the existing schemes for the copyright protection of scalar data in WSNs using digital watermarking and discussed three of these, while the problem of copyright protection itself was discussed in Section 3.2.2. Based on the discussion of this problem, we can identify two technical problems with the existing copyright protection mechanisms as follows:

- Inability to securely add or embed a robust watermark signal to the data sensed by a WSN deployed in a hostile, unattended environment.
 - Inability to make the WSN sensed data robust against severe constraints, such as modification, manipulation, Sybil attack and forwarding attack, during the process of data transmission through the communication channel.
-

These technical problems give rise to a number of research questions to be addressed, in order to develop effective copyright protection of scalar data during communication between sensor nodes.

For example, the following questions need to be answered:

- a. How to add or embed a robust watermark to the WSN sensed data in a hostile, unattended environment.
- b. How to make the WSN sensed data robust against severe malicious attacks, such as modification, manipulation, Sybil attack, and selective forwarding attack to scalar sensor node, during transmission between sensor nodes.

3.3.2 Research Issue 2: Developing Copyright Protection of Images in WMSNs using Watermarking Technique

Chapter 2 (Section 2.7) presented a survey of the existing copyright protection schemes in WMSNs, using digital watermarking. However, the majority of them work on the authentication of the image and the signal. Some of them work on the secure communication of the image. There is only one scheme for copyright protection of images in WSNs, and no such scheme for WMSNs. The problem of copyright protection was discussed in Section 3.2.3. Based on this discussion, three technical problems can be identified with the existing authentication mechanism, as follows:

- Inability to explain when and where the process of embedding and extraction are to be undertaken.
- Inability to add or embed a robust watermark to the multimedia data from WMSNs deployed in hostile, unattended environments.
- Inability to make the WMSN multimedia data robust against severe constraints, such as modification, manipulation, Sybil attack and forwarding attack, during data transmission through the communication channel.

The technical problems outlined above give rise to a number of research questions to be addressed, in order to provide for copyright protection of images transmitted between sensor nodes. For example, the following questions need to be answered:

- a. How to explain when and where the embedding and extraction processes are to be undertaken.
-

- b. How to add or embed a robust watermark to the multimedia data from a WMSN deployed in a hostile, unattended environment.
- c. How to make the WMSN multimedia data robust against severe constraints, such as modification, manipulation, Sybil attack and forwarding attack, during data transmission through the communication channel.

3.4 Research Methodology

A science and engineering-based research approach will be adopted for this research. Science and engineering research leads to the development of new techniques, architecture, methodologies, devices or a set of concepts, which can be combined to form a new theoretical framework. This research approach commonly identifies problems and proposes solutions to these problems.

The science and engineering approach that has been utilised in this research consists of: problem definition, conceptual solution, implementation, experimentation, testing and validation of prototypes against existing solutions. It consists of three main stages:

- Problem definition
- Conceptual solution
- Implementation, testing and evaluation

3.4.1 Problem definition

In the problem definition stage, the aim is to highlight the significance of the research questions. This problem definition stage has been covered in this chapter. Problems secure valuable sensor scalar and the image during communication between sensor nodes by digital watermarking technique in WMSNs has been grouped into two categories. For each category, the discussion was carried out from three perspectives: a formal definition of the category, the socio-economic, and the technical concerns. The problems associated with the technical concerns led to the research issues for the new solution development.

3.4.2 Conceptual solution

Design is one of the most important parts of a system development process. It involves the study and understanding of the domain, the application of subject matter knowledge and experience to solve the problem and the creation and evaluation of a proposed solution.

In this stage, a conceptual framework is designed for the proposed solution. A conceptual framework is an abstract model of the practical solution. It provides a road map for building the actual solution for the system and can be regarded as a blueprint for the implementation of the system.

3.4.3 Implementation, test and evaluation

In this stage, testing and validation are carried out through experimentation with real-world examples and field testing. The process of testing and validating a working system provides unique insights into the benefits of the proposed concepts, frameworks and alternatives.

By building a prototype system, implementing, testing and evaluating, a better insight into the feasibility and functionality of the conceptual framework as well as the whole solution is provided.

3.5 The Objective of the Research

In the previous discussion, we defined two main problem areas, and raise two research issues. In this section, we will discuss the two research objectives for the solution development that address these research issues. The two objectives are:

Objective 1

To develop a watermarking algorithm to address the issue of copyright protection of scalar sensed data in WSNs. The proposed watermarking technique is aimed to robust against various types of attack, such as scalar data modification, scalar data insertion and scalar data repetition. The watermarking technique is satisfy all the watermarking properties i.e. robustness and informed detection.

Objective 2

To develop a watermarking algorithm to address the issue of copyright protection of sensed image in WMSNs. The proposed watermarking technique is aimed to robust against severe constraints on security such as kinds of attacks during the image in communication channel such as replication, modification, and manipulation.

3.6 Summary the problem Definition & Research Issues

In this chapter, we have identified two major research issues that are aimed at solving information security problem associated with copyright protection of scalar data in WSNs and copyright protection of the image in WMSNs. In this section, we will provide a brief summary of each the following research issues:

- Problem with copyright protection of scalar data in WSNs
- Problem with copyright protection of image in WMSNs

Research Issue 1

In the first part of the theses, we intend to develop a watermarking solution that is robust against the duplicate of the scalar sensed data and the severe constraints attack of the sensory scalar data on security such as modification and Sybil attack.

The proposed solution will offer the following features:

- It provides securely add or embed a robust watermark signal to WSN sensed data
 - It provides robust WSN sensed data against severe constraints on security such as kinds of attacks during the image in communication channel such as modification, manipulation, Sybil and forwarding attack.
-

Research Issue 2

In the second part of theses we intend to develop a watermarking solution that is robust against the duplicate of the image during transmission on the image and the severe constraints attack of the sensory scalar data on security such as modification and Sybil attack.

Hence, we intend to develop a watermarking technique which can offer the following features:

- It provides when and where the process of embedding and extracting are undertaken.
- It provides add or embed a robust watermark to WMSN multimedia data which is deployed in hostile unattended environments.
- It provides WMSN multimedia data against the following severe constraints attacks to WSN sensed data such as modification, manipulation, Sybil and forwarding attack during the data transmission through a communication channel.

3.7 Conclusion

This chapter provides a problem definition for securing valuable sensor scalar and the image during communication between sensor nodes in WMSNs by digital watermarking techniques and approaches. Based on the socio-economic and technical problems of existing solutions, two research issues have been defined. For each research issue, a number of research questions have been proposed. These research questions need to be addressed in the development of any new digital watermarking techniques in WMSNs. To address each research issue, two research aims have been proposed. Furthermore, the research methodology for this research has been discussed.

In the next chapter, an overview of the proposed solution along with its conceptual framework will be provided. The conceptual framework is designed to address all the issues that have been discussed in this chapter.

CHAPTER FOUR

AN OVERVIEW OF THE SOLUTION AND THE CONCEPTUAL PROCESS

This chapter presents

- ▶ an overview of the proposed solutions,
- ▶ a conceptual framework for the proposed solutions,
- ▶ the conceptual process adopted in the development of the proposed solutions.

4.1 Introduction

As seen in Chapter 2, a number of works have addressed the issue of secure transmission, authentication, integrity and copyright protection between sensor nodes in WMSNs. However, few of them have been able to resolve the problems in their entirety. Therefore, Chapter 3 presented two major research issues pertaining to the existing solutions proposed in this research. This chapter will have a look at these proposed solutions.

4.2 The Proposed Solution : Overview

Chapter 3 gave the problem definition listing two main problems regarding information security:

- Problems with copyright protection of the scalar data in WSNs using digital watermarking technique.
 - Problems with copyright protection of images in WMSNs using digital watermarking technique.
-

Chapter 3 further identified the research issues emanating from these problems. As it pointed out, in order to provide for copyright protection of scalar data, the solution intended to be developed should have the following features:

- Ability to securely add or embed a robust watermark signal to the data sent by a WSN which is deployed in hostile, unattended environments.
- Ability to make the WSN sensed data robust against the severe constraints on its security, such as modification, manipulation, Sybil attack and forwarding attack, during data transmission through the communication channel.

The proposed watermarking technique (LKR Watermarking technique), as the solution that addresses these issues, is presented in Chapter 5. This approach to copyright protection of scalar data using digital watermarking technique for WSNs addresses the following two issues: to securely add or embed a robust watermark signal to the WSN sensed data, and the ability to make the WSN sensed data robust against the severe constraints on its security. The LFSR and Kolmogorov rule provide for these features, as Figure 4.1 shows.

The next problem is how to protect the copyright of images in WMSNs. In order to address the issue of determining where the process of embedding and extraction are undertaken and how to overcome the malicious attacks, we propose another watermarking technique (GPKR Watermarking technique) for the copyright protection of images. The solution to be developed should have the following features:

- Ability to explain when and where the process of embedding and extraction are to be undertaken.
- Ability to add or embed a robust watermark to the multimedia data from a WMSN deployed in hostile, unattended environments.
- Ability to make the WMSN multimedia data robust against the severe constraints on the security of the sensed data, such as modification, manipulation, Sybil attack and forwarding attack.

The model for copyright protection of images in WMSNs using digital watermarking technique has been proposed in Chapter 6, which does it by reducing and expanding the image using the Gaussian pyramid and Kolmogorov rule, as shown in Figure 4.1.

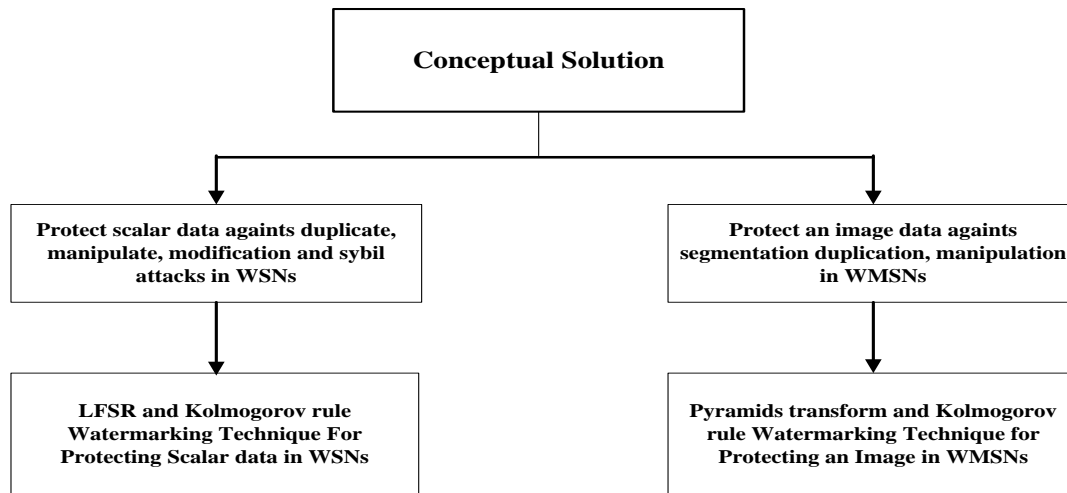


Figure 4.1 An overview of the conceptual solution

The thesis is organized from this chapter onwards as shown in Figure 4.1. The conceptual solutions proposed to address the two main problems of information security have been discussed in this chapter. The next chapter describes these solutions in greater detail.

4.3 Solution Description

This section looks at the solutions developed to resolve the issues discussed in earlier chapters. The two proposed solutions are follows:

- LKR Watermarking Technique: This is a technique to protect copyrights of WSN sensed scalar data using LFSR and Kolmogorov rule.
- GPKR Watermarking Technique: This is a technique to protect copyrights of WMSN sensed images using the Gaussian Pyramids transform and Kolmogorov rule.

Each of these solutions will now be explained in detail, beginning with the watermarking technique using LFSR and Kolmogorov rule for copyright protection of scalar data in WSNs.

4.3.1 LFSR and Kolmogorov Rule for Copyright Protection of Scalar Data in WSNs (LKR Watermarking Technique)

As mentioned in Chapter 2, an algorithm for copyright protection of scalar data should satisfy the following properties:

- Embedding effectiveness
- Imperceptibility
- Informed detection
- Robustness

The watermarking solution (LKR Watermarking technique) proposed in this thesis satisfies these properties. To generate the cover medium, we introduce atomic trilateration which was used for generating the cover medium in section 2.1.1. The sensed scalar data, i.e. the phenomena from the external environment captured by the sensor nodes, is protected against severe malicious attacks by embedding a watermark signal. Watermark signal is a kind of signal or pattern that can be inserted into the cover medium. It is generated by LFSR by converting the scalar data and the message sensed data into binary sequence, i.e. the information that is conveyed by the watermark signal. This message sensed data is used to communicate from the sender to the receiver. The watermark constraints are generated by the watermark signal incorporating the Kolmogorov rule. To test the embedding effectiveness, the LKR watermarking technique embeds the message sensed data and the watermark constraints into the cover medium and immediately detects their presence before applying any attacks. The LKR watermarking technique operates in the spatial domain. The imperceptibility of the watermarked cover medium is maintained by embedding the watermark constraints into the cover medium and immediately detecting their presence before applying any attacks. Non-blind detection and robustness is achieved by measuring the normalized difference error between the watermarked solution and the solution obtained without watermark. This algorithm is robust against the following attacks: deletion, packet replication, and Sybil attack. Figure 4.2 shows how LKR watermarking technique works for the copyright protection of scalar data in WSNs.

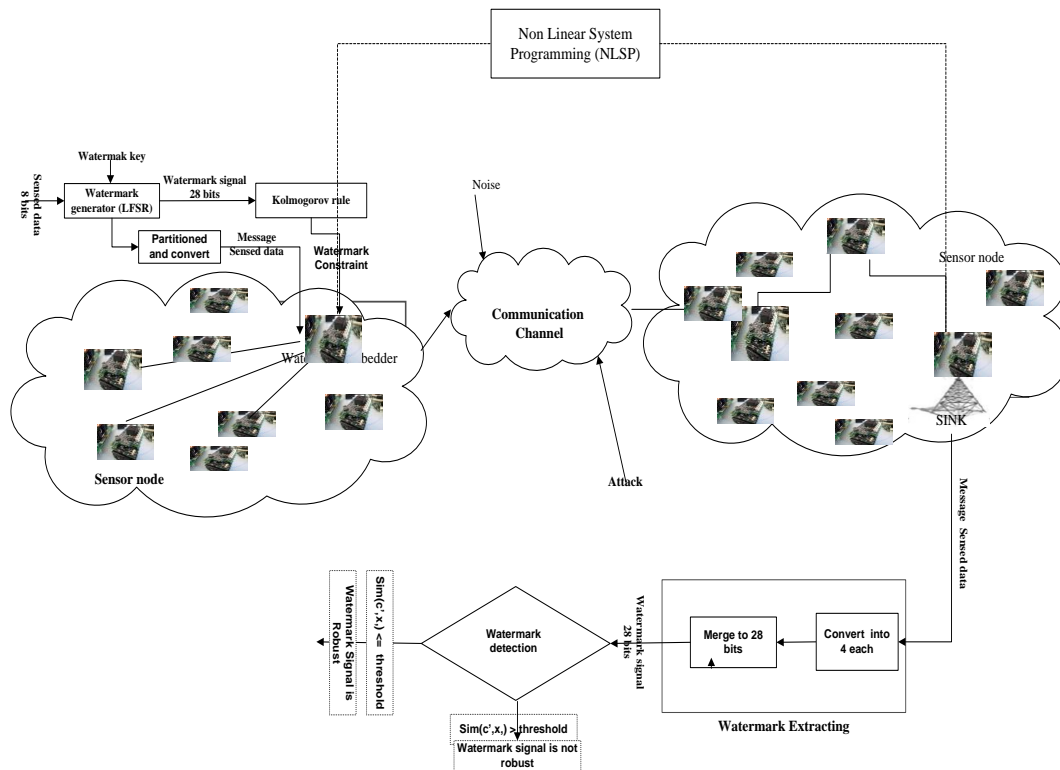


Figure 4.2 Working LKR Watermarking technique for copyright protection of scalar data in WSNs

After this discussion on the LKR watermarking technique in this section, the next section explains the GPKR watermarking technique for copyright protection of images in WMSNs.

4.3.2 Gaussian Pyramids and Kolmogorov Rule for Copyright Protection of Images in WMSNs (GPKR Watermarking Technique)

As mentioned in Chapter 2, an algorithm for the copyright protection of images should satisfy the following properties:

- Embedding effectiveness
- Imperceptibility
- Informed detection
- Robustness

The watermarking solution (GPKR Watermarking technique) proposed in this thesis satisfies these properties. Atomic trilateration was used to generate the cover medium in Section 2.1.1. However, WMSNs deal with more complex data gathered from the targeted area, e.g. images, and audio and video streaming. Therefore, a WMSN has the image as the main sensor. This image refers to the sensed image by a multimedia sensor node and is then reduced by the Gaussian pyramid transforms to become the reduced image. The Gaussian pyramid transform is a technique used for reducing and extending images, as described in section 2.2.4. The reduced image is converted into sequences of binary streams, called watermark signals. The watermark constraints are obtained by generating these watermark signals, incorporating the Kolmogorov rule. To test the embedding effectiveness, the GPKR watermarking technique embeds watermark constraints into the cover medium, and immediately detects its presence before applying any attacks. The GPKR watermarking technique operates in the spatial domains of the images. The imperceptibility of the watermarked cover medium is maintained by embedding watermark constraints. Non-blind detection and robustness is achieved by measuring the normalized difference error between the watermarked solution and the solution obtained without watermark. This algorithm is robust against insertion and packet replication attacks. Figure 4.3 shows how GPKR watermarking technique works for the copyright protection of images in WMSNs.

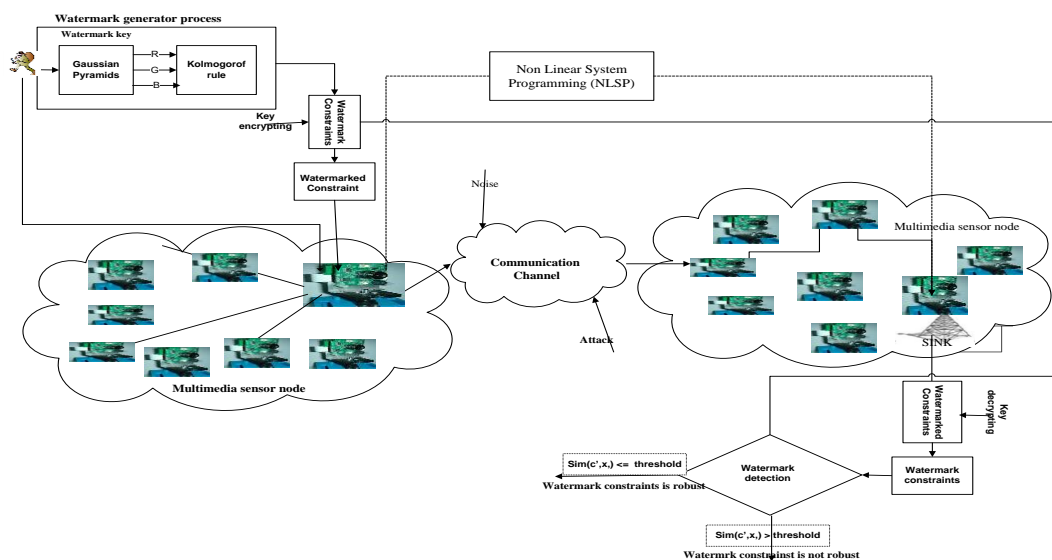


Figure 4.3 Working GPKR Watermarking technique for copyright protection of images in

WMSNs

This section explained the GPKR watermarking technique for the copyright protection of images in WMSNs. The next section describes the approach adopted in this research, explaining the conceptual process followed in the solution development. As mentioned earlier, the research follows a science- and engineering-based approach, and system design and development methodology which is discussed next.

4.4 Conceptual process

The overview of the proposed solution has been discussed in Section 4.2. However, developing the solution requires strictly following a conceptual process. This involves clearly eliciting the requirements and enunciating the design rationale, proposing a theoretical foundation that addresses the requirements, building a prototype system that can verify the theoretical foundation, and finally validating and verifying the solution.

This section outlines the conceptual process followed consistently throughout the entire research that includes two projects covered by this thesis. The main steps of the conceptual process are illustrated in Figure 4.4.

As already mentioned, the research adopts a science- and engineering-based research approach and follows the system design and development methodology (T.M. Salvatore 1995), in order to develop a technologically advanced solution, as illustrated in Figure 4.4

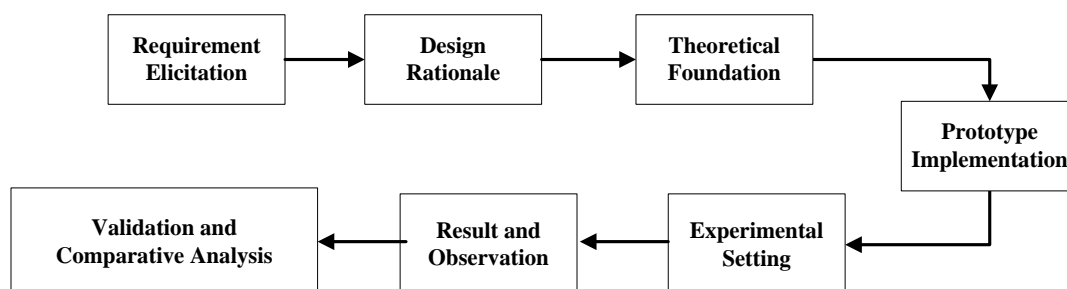


Figure 4.4 Conceptual process followed in this thesis

Each of these main stages will now be explained in detail, listing the activities carried out at each stage.

4.4.1 Requirements, Elicitation and Prioritization

This stage begins with a survey of the existing solutions, and identification of the main issues relevant to these solutions. These issues are then prioritised according to their importance. For example, the issues in the copyright protection of scalar data in WSNs are: how to overcome the duplication of the scalar sensory data during transmission, and how to handle the severe malicious attacks during the transmission of scalar data. Similarly, the issues in the copyright protection of images in WMSNs are: how can the process of embedding and extraction be made clear, and how can it be determined as to when these processes are to be undertaken? The issues higher up on the list are given a high priority while designing the solution, while it offers the flexibility to address other issues if desired.

4.4.2 Design rationale

The second stage in the conceptual process is design rationale. Once the requirements elicitation and prioritization stage is finished, the next stage (i.e. design rationale) describes the basic design decisions to meet the requirement elicited in Stage 1. Each and every listed requirement contributes to the design process. We relate each and every requirement to a design, thereby that all the requirements are considered when finalizing the design process.

4.4.3 Theoretical foundation

The stage of theoretical foundation emanates from the design rationale. It proposes a solution to address all the requirements listed in Stage 1 by analysing the design rationale determined in Stage 2. A detailed algorithm is proposed to cover all the needs.

The feasibility, computational complexity, scalability, real time deployment, and reliability of each design decision are investigated and the final algorithm incorporates all the design decisions that meet these criteria.

The final algorithm needs to be tested for the functionality purposes and, later on, the results of the tests can be used as insights to optimise the initial algorithm. This test implementation is carried out in the next stage of prototype implementation, as described below.

4.4.4 Prototype Implementation

This stage provides for an implementation of the theoretical foundation and prototype algorithm, designed in any development environment that suits the needs. For instance, watermarking technique can be implemented using MATLAB and TOMLAB, development platforms which have been used in a large number of scientific projects requiring a high degree of precision, for example, precision up to 8 decimal points. Further, these are quite stable environments offering many built-in functions that remove the necessity of testing the basic functions (e.g. reducing of the image or expanding the image, and converting decimal to binary or binary to decimal), allowing the developer to focus on algorithm development. Experiments and tests are performed on the prototype and observations recorded to validate the theoretical foundation. However, before carrying out experiments on the prototype, it is necessary to identify the test cases, thereby finalising the experimental setting which is the next stage explained below.

4.4.5 Experimental Setting

At this stage, the experiments run on the prototype are discussed to analyse its performance in different conditions in the real world situations. Since each experiment is different from others, this stage finalizes the experimental settings, defining the minutest details, like the coordinate position of the two-dimensional sensor networks using the random position of a sensor node, the scalar sensed data, and the vulnerability to the attacks, to ensure that the results obtained from the prototype are reliably and consistently estimated.

The experiments are carried out over a limited scalar data pool. In case of watermarking technique, at least 32 two-dimensional positions using random 50 and 75 positions of the sensor node are tested in order to get reliable results. The sensed scalar data and images vary a lot in their position in the tests. After the experiments are over, tabulation of results is done for further analysing and improving the prototype, as explained below.

4.4.1 Results and Observations

This stage is concerned with analysing results and observations. At this stage, the experiments run on the prototype are discussed and observations described. In case of watermarking, the watermarked cover medium is attacked by several types of attacks, such as deletion, replication or

modification of data, insertion of false data, Sybil attack and selective forwarding attack. After each attack, the watermark constraints are extracted from the watermarked cover medium to check for their presence, in order to identify the watermark constraints that are still robust against the attack. The results are then validated by cross-relating the observations with the theoretical foundation, as described below.

4.4.2 Validation and Comparative Analysis

In this final stage, the actual results (results from the experiments) are compared with the expected results (results from the theoretical model), which might be similar to, or quite different from, the actual results. In case of differences, the theoretical model is adjusted and all the steps from stages 1 to 6 are amended. After the validation of the expected results, the proposed solution is compared with the existing solutions to discover its strong and weak points. This comparative analysis provides new avenues for future research.

4.5 Conclusion

In Chapter 2, a survey of the literature on watermarking techniques for WSNs and WMSNs was carried out. Based on the survey, in Chapter 3, the existing issues in the field were identified and outlined. It was also noted that, although several watermarking techniques have been proposed in the literature, none of them addresses all the issues outlined.

Therefore, two watermarking techniques, termed copyright protection of scalar data in WSNs (LKR watermarking technique) and copyright protection of images in WMSNs (GPKR watermarking technique) were proposed as possible solutions.

Further, in Chapter 3, it was shown how each of these solutions addresses the issues identified. The LFSR and Kolmogorov rule were applied to address the issue of copyright protection in WSNs, and the Gaussian pyramids and Kolmogorov rule were applied to address the issue of copyright protection of images in WMSNs.

CHAPTER FIVE

LKR WATERMARKING TECHNIQUE

This chapter presents:

- ▶ an introduction to the LKR Watermarking Technique for copyright protection of scalar data in WSNs,
- ▶ a general overview of the proposed LKR watermarking technique as the proposed solution for copyright protection of scalar data in WSNs,
- ▶ experimentation and testing of the proposed LKR watermarking technique for copyright protection of scalar data in WSNs,
- ▶ evaluation, validation and a comparative study of the proposed LKR watermarking technique for copyright protection of scalar data in WSNs.

5.1 Introduction

This chapter addresses the problem of copyright protection of scalar data by ensuring that the proprietary information (i.e., the watermark) remains secure between the sensor nodes in a WSN. It particularly considers the issue of copyright protection of scalar data for applications like intellectual property protection. It presents a novel LFSR (Linear Feedback Shift Register) and Kolmogorov Rule (KR) based watermarking scheme, termed LKR, that embeds the message sensed data and watermark constraints into the cover medium (NLSP). What sets this watermarking scheme apart from other such schemes is the use of message sensed data and watermark constraints derived from the sensed data by LFSR and Kolmogorov Rule, leading to its nomenclature as LKR watermarking technique.

5.2 Proposed LKR Watermarking Technique

This section provides a general overview of the proposed watermarking technique, and then outlines the requirements to address the problem.

5.2.1 General overview of LKR

The primary functions of a Wireless Sensor Network (WSN) are collecting and storing data, and forwarding them to other nodes and/or servers. Current technologies allow validation of data during transit, but not after the data has reached its destination. One of the challenges with these technologies is to ensure that the source of the data is preserved, once it has left the WSN. This is important as the data can be used by other applications or distributed to other parties without the consent of the owner. Therefore, it needs to be ensured that the data source is identifiable. However, data sensed by a WSN can be deleted or modified, or false data can be inserted into it, by malicious attackers. Sensors can even be physically captured and replaced by them. Hence, security becomes an important issue with WSNs. The proposed watermarking technique aims to ensure that the copyright information is securely embedded within the source data, so that the owner of the data can be identified, if required.

5.2.2 Requirements

To tackle the issue of copyright infringement of valuable scalar sensor data, the following requirements have been laid down for the proposed LKR watermarking technique. This represents the first stage of the conceptual process described in section 5.4 where the requirements are elicited and prioritized. Although the proposed algorithm has been implemented for scalar sensor data, it can be further extended to images and audio. The requirements are as follows:

1. Coverage: The LKR algorithm should be able to operate on a minimum of 75 nodes over a 500 square meter area.
 2. Cover Medium: The LKR should only have scalar data for the cover medium.
 3. Copyright protection: The LKR should be able to securely embed the copyright information into the cover medium.
-

4. Message sensed data: The LKR should use message sensed data because the message sensed data can provide subjective extraction. The message sensed data is a signal or pattern which is embedded into the cover medium.
5. Watermark constraints: The LKR should use a watermark constraint rather than a binary watermark because the watermark constraints can provide subjective detection.
6. Robustness: The watermark should be embedded robustly to handle watermark attacks.
7. Informed detection: The LKR should offer informed extraction and detection.

Based on the requirements outlined above, Section 5.2.3 discusses the design rationale of the LKR watermarking technique, incorporating each of these requirements in the design rationale. It also shows how the design decision addresses each requirement. This discussion concludes stage one of the conceptual process and paves the way for the next stage which is design rationale.

5.2.3 Design Rationale

This stage represents the second step of the conceptual process referred to in Section 4.4.2, in which all the requirements are addressed by finalising the design decision. The theoretical foundation of the proposed LKR watermarking technique satisfies the requirements outlined in Section 5.2.2, as illustrated in Figure 5.1. The following design decisions are proposed in the framework

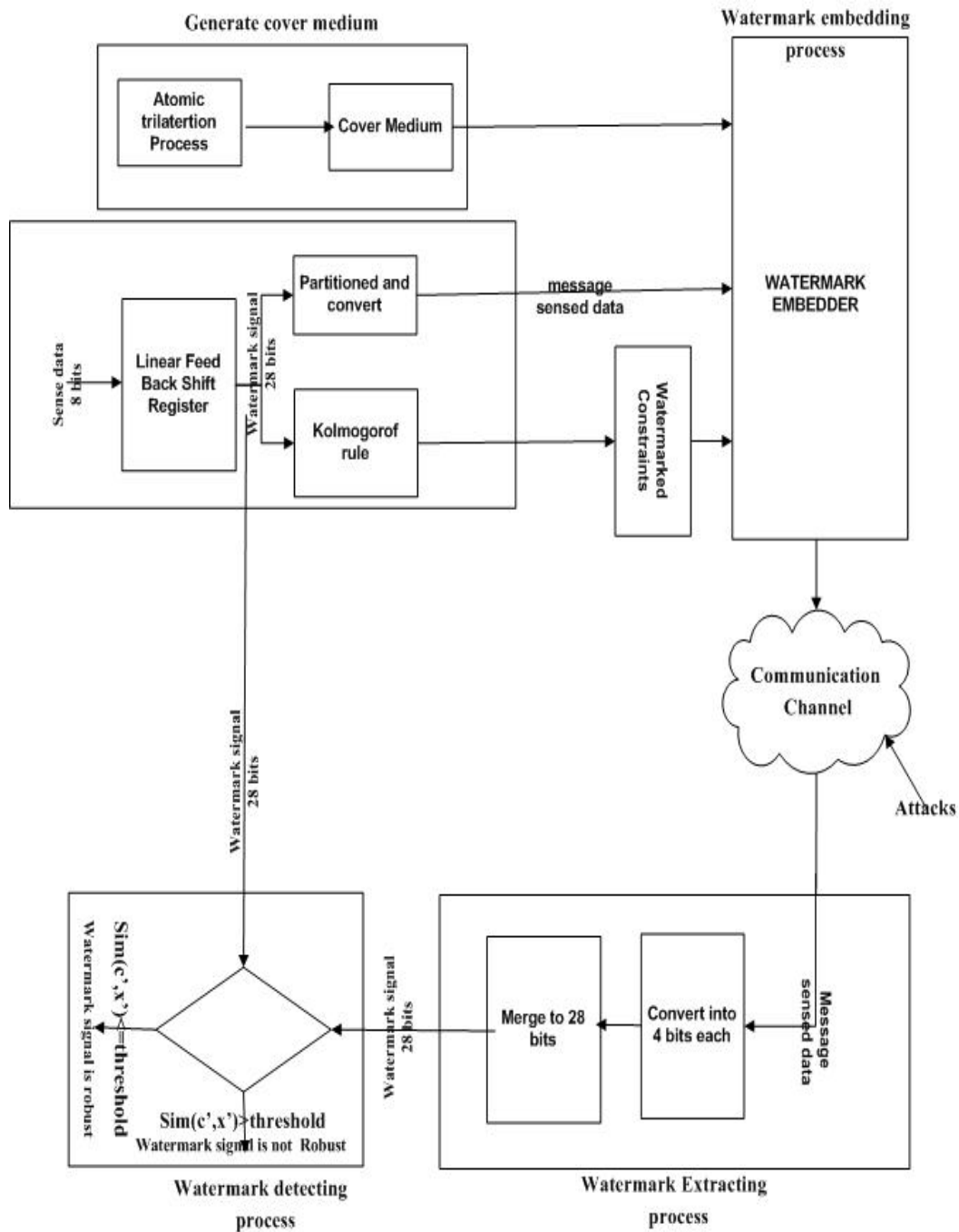


Figure 5.1 Theoretical foundation of LKR watermarking technique in WSNs

1. The network setting consists of 75 sensor nodes placed randomly within a 500 square meter area. This network provides random positions to three sensor nodes that are used to start up the cover medium generation process (Req. 1).
2. The cover medium is generated by using the theory of atomic trilateration, as discussed in Section 0. This cover medium is used to embed the watermark constraints and message sensed data (Req. 2, Req. 3).

3. The message sensed data is originally generated using the scalar data, and then this scalar data is converted to 8 binary bits, which are further expanded to 28 bits by LFSR, and the message sensed data results from partitioning it to 4 binary bits, which are then converted into decimals (Req. 4). Figure 5.2 shows how the message sensed data is created:

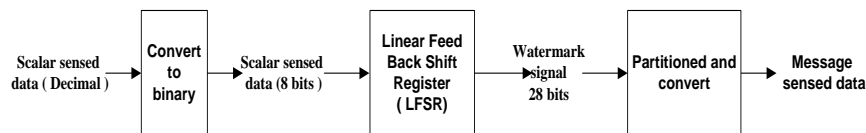


Figure 5.2 The process of the creation of message sensed data

4. Watermark constraints are generated from the original scalar data captured by the WSN node. The scalar data is converted into 8 binary bits, these bits are then expanded to 28 binary bits by LFSR and, using the Kolmogorov rule, the watermark constraints are created, as illustrated in Figure 5.1 (Req. 5)).
5. Robustness of the watermark constraints is measured using the normalized difference error between the watermarked solution and the solution obtained without watermark (Req. 6).
6. The cover medium is required for watermark signal detection because the watermark signal can produce message sensed data, and then the message sensed data is converted and merged to get the sensed data (Req. 7).

5.3 Theoretical Foundation for LKR Watermarking Technique

This section proposes the theoretical foundation for copyright protection of scalar data. This represents the third stage of the conceptual process described in Section 4.4.3 where the design decisions are analysed and algorithm is presented. The proposed scheme offers robustness against data deletion, packet replication and Sybil attack, which are not tackled in the literature (Zhang, Liu, and Das 2008) (Kamel 2011). Compared to this scheme, Zhang, Liu, and Das (2008) have put forward a statistical watermark approach to authenticate data which is end-to-end and inherently supports in-network processing. In this technique, watermark is actually the modulated

authentication information, embedded in the sensory data at sensor nodes. This technique resists attacks like false distribution or imposition on the part of the sensor nodes, remnant check in the spatial domain, coefficient value forgery, and non-zero coefficient position switch in the frequency domain. This technique is robust against false distribution on the part of the sensor nodes and remnant attack in the spatial domain. However, although it can detect coefficient value forgery and non-zero coefficient position switch in the frequency domain, the authors do not discuss whether this technique is robust against these or not. A communication protocol for WSNs has been introduced by Xuejun (2010) to authenticate sensitive data transmission. The technique uses sensitive information as the watermark. The watermark is then embedded in the sensory data in the sensor nodes. A threshold is used to avoid the alteration of the lowest bit "1", append "1" into the Output Binary system (OBS), or otherwise append "0" to it, and make a big influence on the precision of the sensory data. However the author does not discuss what kind of attacks have been used and whether the technique is robust against attacks. Kamel and Juma (2011) have introduced a technique to provide for data integrity. This technique, based on distortion free watermarking, embeds the watermark in the order of the data element, so that it does not cause distortion of the data. This technique uses "man-in-the-middle attack" as the vulnerability attack, and other attacks, such as modification of data or insertion of false data. It is robust against modification and false data attacks. Our scheme embeds watermark constraints and the message sensed data in the cover medium. Our watermark constraints are generated by using the Kolmogorov rule and the message sensed data, by partitioning and converting them into decimals. Both watermark constraints and the message sensed data are added to the cover medium only.

The novel feature of the LKR watermarking technique is that it uses the LFSR and Kolmogorov Rule for copyright protection of scalar data. The LFSR is used to generate the watermark signal, and the Kolmogorov rule is used to generate the watermark constraints. The LKR watermarking comprises four steps:

- Cover medium generation
 - Watermark generation
 - Watermark embedding
 - Watermark extraction & detection.
-

5.3.1 Cover Medium Generation

This section explains the process of generating the cover medium by using the atomic trilateration process. The pseudo-code for generating the cover medium is shown in Pseudo Code 5.1 and the flowchart in Figure 5.3.

Pseudo Code 5.1 Generating cover medium

Input:

$(x_A, y_A), (x_B, y_B), (x_C, y_C)$	Position of two-dimensional three sensor network
T_c	Temperature of the propagation media
t_{DA}, t_{DB}, t_{DC}	Time for transmission between node D to A, D to B and D to C
V_s	Speed of the acoustic signal
ϵ_t	Error in the measurement of the temperature
$\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$	Error in the measurement of the timer from D to A , D to B and D to C
$\delta_1, \delta_2, \delta_3$	Error in the measurement between the Euclidean measurement and the measurement using time differences of optimal D to A, D to B and D to C.

Output

The cover medium

$$\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3$$

Constraints

$$\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3$$

Step 1: Getting all input Data

Position of the two-dimensional, three-sensor network $(x_A, y_A), (x_B, y_B)$, and (x_C, y_C) . Temperature of the propagation media (T_c), Time for transmission between node D to A, D to B, and D to C , Error in the measurement of the temperature (t_{DA}, t_{DB}, t_{DC}), Error in measurement of the timer from D to A , D to B, and D to C ($\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$), and Error in measurement between the Euclidean measurement and the measurement using time differences of the optimal D to A, D to B, and D to C ($\delta_1, \delta_2, \delta_3$).

Step 2: Computing the speed of the acoustic signal and adding the error of temperature, because this speed is one of the requirements for measuring the distance between two sensor nodes.

Compute the speed of the acoustic signal using $V_s = 331.4 + 0.6(T_c + \epsilon_t)$

Step 3: Computing the distance between the node D and the sensor nodes A, B and C, using time differences of arrival (TDoA) and adding the error of measurement of the timer respectively.

Compute the distance of $d_{DA} = V_s * (t_{DA} + \epsilon_{DA})$, $d_{DB} = V_s * (t_{DB} + \epsilon_{DB})$, and $d_{DC} = V_s * (t_{DC} + \epsilon_{DC})$.

Step 4: Computing the distance between the node D and the sensor nodes A, B and C, using the Euclidean theorem.

Compute the distance of $d_{DA} = \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}$, $d_{DB} = \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2}$, and $d_{DC} = \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}$

Step 5: Computing the difference between the distance using (TDoA) and the distance using the Euclidean theorem, and adding the error in measurement between the Euclidean measurement and the measurement using time differences of the optimal D to A, D to B, and D to C.

Use step (2) to step (3) to compute the difference between step (4) and (3) and add the error between step (4) and step (3).

$$\begin{aligned} \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) &\leq \delta_1 \\ \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) &\leq \delta_2 \\ \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) &\leq \delta_3 \end{aligned}$$

Step 6: Computing and minimizing the errors in the system of equations (step 5).

Compute and minimize the objective function $\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3$

Step 7: Generating Cover medium

Append step 5 and step 6 to get a Non-Linear System Programming

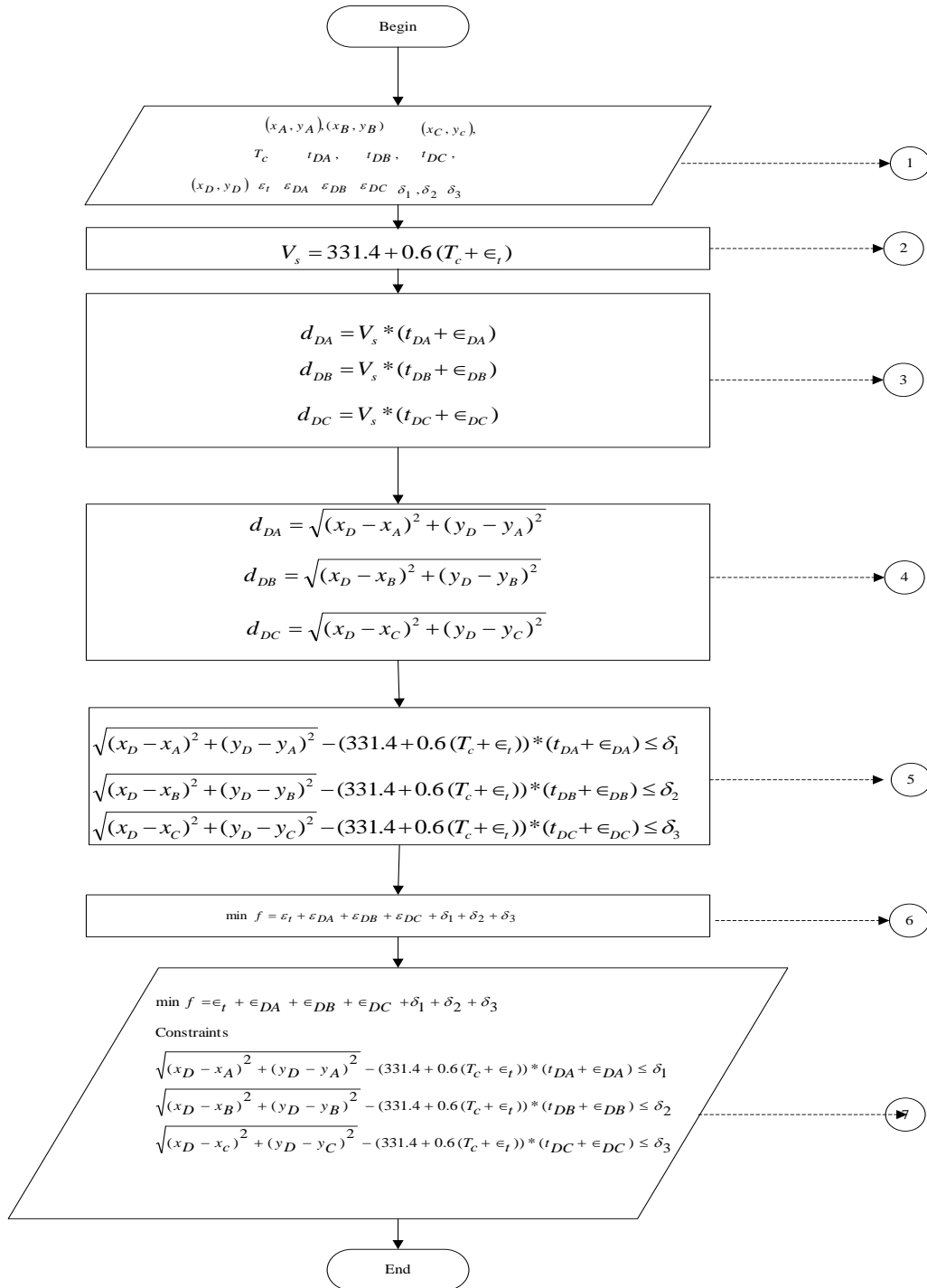


Figure 5.3 Flowchart of the generation of Cover Medium

5.3.2 Watermark Generation

This subsection explains the process of generation of the watermark signal. The generation process is the first and a very crucial step in the process, with unique and complex requirements. The information contained in the watermark message, such as text and/or sensed data, must be unique

so that it can be used for ownership identification. Both the watermark message and the watermark key generator are inputs, processed in the watermark generator to produce a watermark signal. The process of generating watermark signal consists of four steps:

1. Converting sensitive data into binary sequences,
2. Using LFSR to create watermark signal,
3. Producing watermark constraints using Kolmogorov rule
4. Partitioning and converting the watermark signal to decimal numbers in order to produce the message sensed data.

Each of these steps will now be described in detail.

5.3.2.1 Converting sensitive data into binary

The first step involves the conversion of the sensitive data into binary sequences. The term sensitive data refers to the information which is to be protected to prevent undesirable disclosures, restricting access to it to authorized agencies only. The reasons to protect such information may be legal or ethical, e.g., personal privacy. For example, WSNs telemonitor the patients' physiological data and track doctors and/or nurses in the hospital in many healthcare applications (Kahn, Katz, and Pister 1999, Noury, Mercier, and Porcheron 2000). In these applications, small sensor nodes are attached to the patients with varying functions. For example, one sensor node may monitor the heart rate while the other detects any abnormalities in blood pressure. The pseudo-code for converting sensitive data is shown in Pseudo Code 5.2 and the flowchart in Figure 5.4.

Pseudo Code 5.2 Converting sensitive data into binary sequences

Input:

d	Sensitive data
---	----------------

Output

B	Binary sequence
Q	Quotient
R	Remainder

Step 1: Getting input Data

d is the sensitive data

Step 2: Dividing data sensitivity by 2 to get quotient and remainder

Sensitive data d is divided by 2

Step 3: Recording quotient and remainder

Record quotient (Q) and Remainder (R) from step 1

Step 4: Conditional

If remainder $Q = 0$ then go to 5, else go to step 1.

Step 5: Collecting remainder R

Collect R s into desired binary number with the first R as LSB and last as MSB

Step 6: Collecting remainder and concatenate

$B = \text{concatenation}(R)$

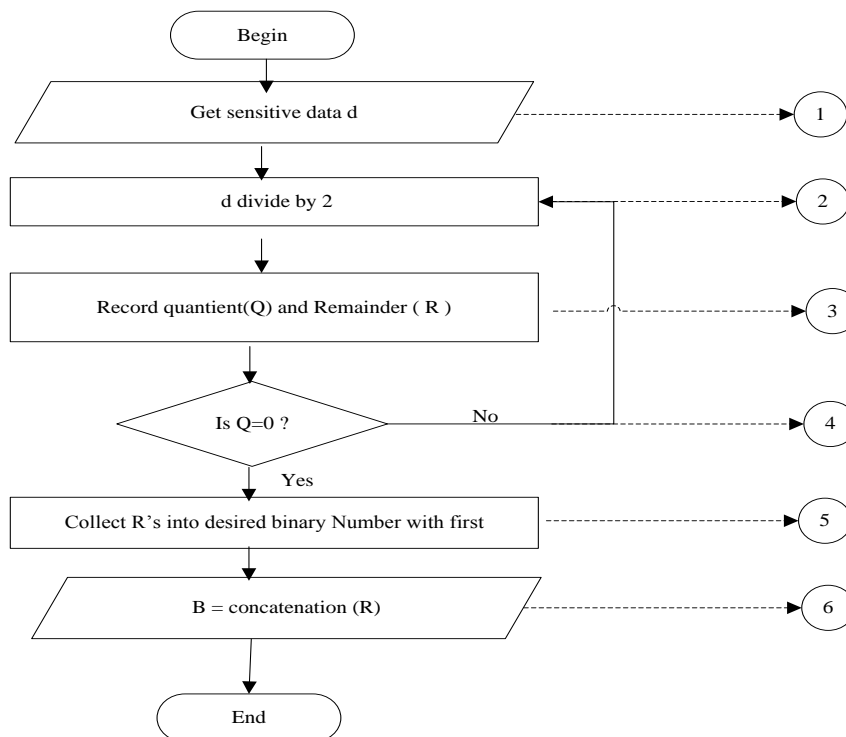


Figure 5.4 Flowchart for conversion of decimal data into binary digits

5.3.2.2 LFSR to create watermark signal

The second step is to generate the watermark signal by using LFSR. The term LFSR refers to a shift register with its input bit as a linear function of its own previous state. Exclusive-or (XOR) is one of the most widely used linear functions of single bits. Thus, most often, LFSR is a shift register with its input bit driven by the XOR of some of the bits from the overall shift register value. Details about LFSR can be found in Section 2.2.2.

The pseudo-code for generating watermark signal is shown in Pseudo Code 5.3 and the flowchart in Figure 5.5.

Pseudo Code 5.3 Generating watermark signal

Input

B	Binary sequence from pseudo code 5.2
$c_i, i = 0, 1, 2, \dots, k$	Coefficient Feedback constants of polynomial $f(x) = c_0 + c_1x + \dots + c_{k-1}x^{k-1} + x^k$ as Watermark key coefficients

Output

W	Watermark signal 28 bit
---	-------------------------

Step 1: Getting input Data

B is the binary sequence produced using pseudo code 5.2

Step 2: Applying initial value as seed

Use binary sequence B as the initial value, B is called the seed

Step 3: Applying watermark key

Use $c_i, i = 0, 1, 2, \dots, k$ as feedback constants of polynomial

$f(x) = c_0 + c_1x + \dots + c_{k-1}x^{k-1} + x^k$ as watermark key coefficients

Step 4: Generating an infinite binary sequence

Generate an infinite binary sequence with B as seed and c_i as watermark key by using LFSR.

Step 5: Cutting the infinite binary sequence

Cut the infinite binary sequence from 1 to 28

Step 6: Producing watermark signal (W)

Cut the infinite binary sequence from 1 to 28

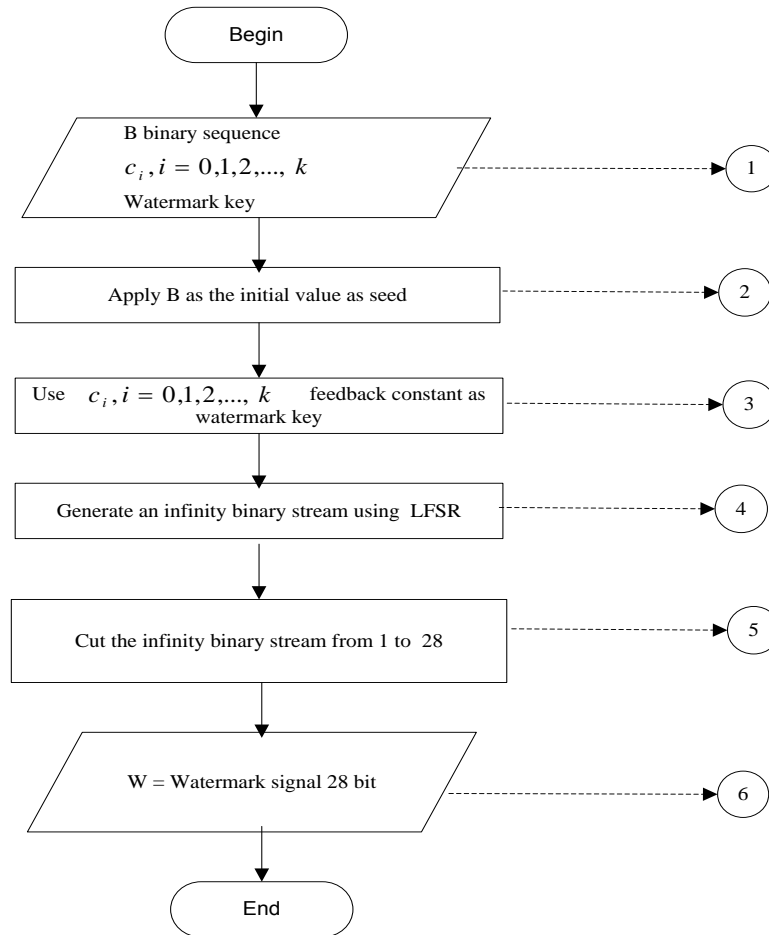


Figure 5.5 Flowchart for generation of watermark signal

5.3.2.3 Kolmogorov Rule to Create Watermark Constraints

This subsection explains how to create watermark constraints using Kolmogorov rule, already introduced in Section 2.2.3. The rule used here can be seen in Table 5.1

Table 5.1 The particular of the Kolmogorov rule

1	2	3	4	5	6	7
\mathcal{E}_t	\mathcal{E}_{DA}	\mathcal{E}_{DB}	\mathcal{E}_{DC}	δ_1	δ_2	δ_3

The pseudo-code for generating watermark constraints is shown in Pseudo Code 5.4 and the flowchart in Figure 5.6

Pseudo Code 5.4 Generating watermark constraints

Input

W	Watermark signal 28 bit
---	-------------------------

Output

WC	Watermark constraint
----	----------------------

Step 1: Getting input Data

Watermark signal 28 bits

Step 2: Grouping watermark signals

Group 28 bits watermark signals into groups of 7 bits each

Step 3: Matching, using Table 5.1

Match the bit number with corresponding variable number using Kolmogorov rule (Table 5.1)

Step 4: Deciding variable included or not included

If a bit is assigned a variable within a group and that variable is included in the linear, go to step 4. Else, if a bit zero is assigned to a variable within a group and that variable is not included in the linear, then go to step 2.

Step 5: Generating watermark constraints

Print WC watermark constraint

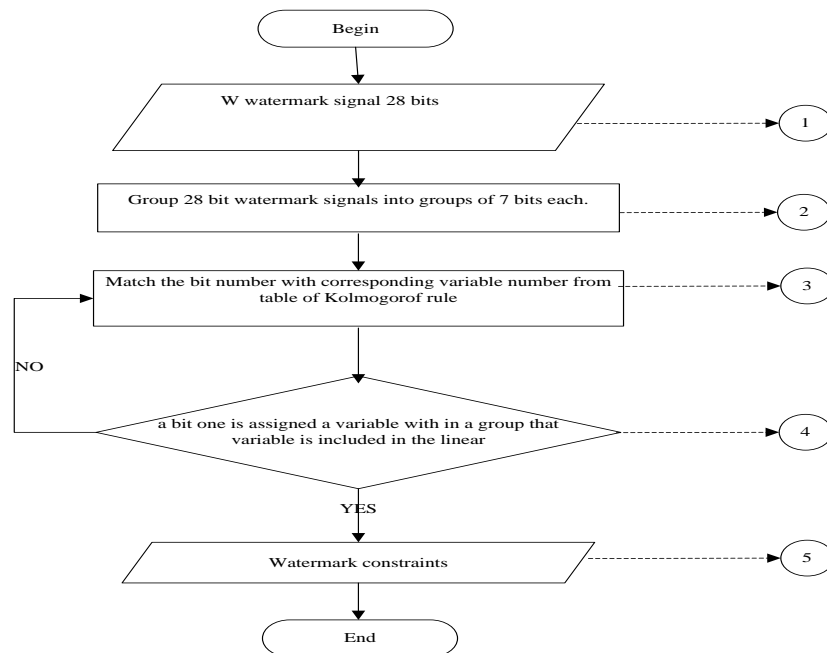


Figure 5.6 Flowchart for the generation of watermark constraints

5.3.2.4 Partition and convert to create message sensed data

This section explains the process of creating the message sensed data. This is actually a sensed data that comes from a 28 bit watermark binary, resulting from the generation of the watermark signal. The pseudo-code for generating the message sensed data is shown in Pseudo Code 5.5 and the flowchart in Figure 5.7.

Pseudo Code 5.5 Generating message sensed data

Input

W	Watermark signal 28 bit
---	-------------------------

Output

MSD	Message sensed data ($Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$)
-----	--

Step 1: Getting input data

Watermark signal 28 bits

Step 2: Dividing 28 bit watermark signal

Divide 28 bit watermark signal into groups of 4 bits each.

Step 3: Converting each group

Convert each of the groups into decimal numbers

Step 4 : Generating message sensed data

Message sensed data ($Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$)

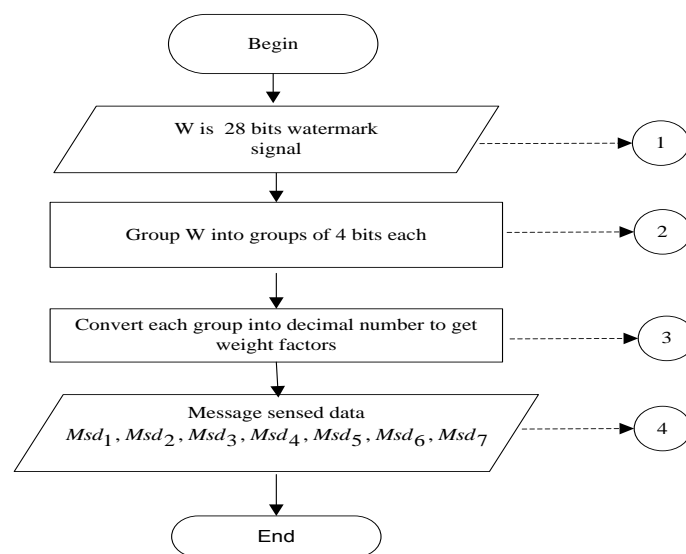


Figure 5.7 Flowchart for the generation of message sensed data

5.3.3 Watermark Embedding Algorithm

This sub section explains the process of embedding the message sensed data and the watermark constraints. The process of embedding forms the second stage in the watermark system. This process is undertaken by an embedder which combines the cover medium, the watermark constraints and the message sensed data. It, then, creates a watermarked cover medium which is perceptibly identical to the cover medium.

The pseudo-code for embedding the message sensed data and the watermark constraints is shown in Pseudo Code 5.6, and Figure 5.8 shows the watermark embedding process

Pseudo Code 5.6 Embedding watermark constraints

Input

WC	Watermark constraints
MSD	Message sensed data
$(x_A, y_A), (x_B, y_B), (x_C, y_C)$	Position of two-dimensional sensor networks
T_c	Temperature of the propagation media
$\tau_1, \tau_2, \tau_3, \tau_4$	The values are selected such that the feasibility of the solution space of the optimization problem is not harmed
t_{DA}, t_{DB}, t_{DC}	Time transmission between node D to A, D to B and D to C

Output

(x_D, y_D)	Position of two-dimensional sensor networks
$\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$	The error in the measurement of the timer from D to A, D to B and D to C
$\delta_1, \delta_2, \delta_3$	The error in the measurement between the Euclidean measurement and the measured using time differences of optimal D to A, D to B and D to C.
ϵ_t	The error in the measurement of the temperature
f	Minimum of the function of objective

Step 1: Getting all input data

The three positions $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ are obtained randomly to form networks of 75 positions of two-dimensional sensor nodes. The temperature of the propagation media (T_c) is obtained by using uniform distribution on interval $[0,75]$. Time transmission of t_{DA}, t_{DB} , and t_{DC} are obtained by using gauss distribution on interval $[0,1]$. The values of τ_1, τ_2, τ_3 and τ_4 are also obtained using gauss distribution on interval $[0,1]$. The watermark constraints (WC) are obtained by applying the pseudo code 5.5.

Message sensed data (MSD) are obtained by applying the pseudo code 5.6.

Step 2: Using the cover medium

Obtain cover medium by applying the pseudo code 5.1. This cover medium is called NLSP which has an objective function, consisting of coefficient objectives.

Step 3: Changing the coefficient objectives to the message sensed data

Change each of the coefficient objectives to the message sensed data respectively.

Step 4: Appending watermark constraints (WC)

Append watermark constraints to the cover medium

Step 5: Computing the new cover medium to solve a new coordinate position, the error in in the measurement of the timer, in the measurement of distance, in in the measurement of the temperature and Minimum of the function of objective.

Compute a new cover medium in which the message sensed data and the watermark constrains are added. TOMLAB is used to solve it, and then $(x_D, y_D), \epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3$ and $\min f$ are obtained.



Figure 5.8 Flowchart for watermark embedding algorithm

5.3.4 Watermark Extraction & Detection Algorithm

The last stage in copyright protection system through watermarks is the process of extraction and detection which is a crucial part because the sender identifies and provide information to the

intended receiver using it. The process of detection and extraction is undertaken by a detector. The detection process uses an extraction unit to first extract the watermark signal, and later compare it with the cover medium. The extraction process consists of two parts, watermark location and watermark information recovery. Detection can be of two types based on the requirement of cover medium, or otherwise, in the detection process. If cover medium is required, it is informed detection; if not, it is blind detection. The LKR watermarking technique uses informed detection to detect the watermark.

5.3.4.1 Watermark Extraction Process

The extraction process is also undertaken in the watermark detector as the message sensed data has to be recovered from the cover medium. The pseudo-code for extracting the watermark signal is shown in Pseudo Code 5.7 and the process of watermark extraction in Figure 5.9.

Pseudo Code 5.7 The process of extracting watermark signal

Input

MSD	Message sensed data ($Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$)
-----	--

Output

W	Watermark signal 28 bit
---	-------------------------

Step 1: Getting all input data

Get the message sensed data ($Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$)

as shown in the coefficient objective function of the cover medium

Step 2: Converting the coefficient objective

Convert the message sensed data ($Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$)

into 4 bits respectively

Step 3: Merging all these 4 bits

Merge all of these 4 bits respectively to make 28 bits

Step 4: Generating watermark signal

Generate watermark signal of 28 bits

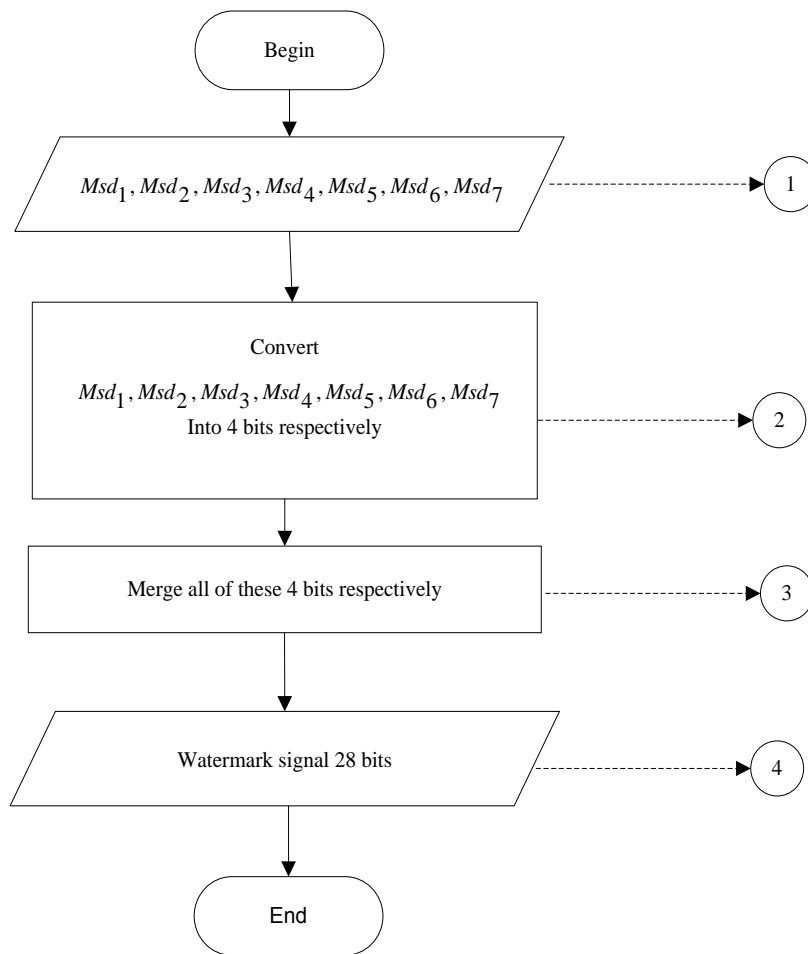


Figure 5.9 Flowchart for extraction of message sensed data

5.3.4.2 Watermark Detection Process

The end of the watermark system is the detection process, which is a crucial part as it allows the sender to identify and provide information to the intended receiver. The detection process is undertaken by a detector. Based on the requirement of the cover medium, or otherwise, detection can be of two types. If a cover medium is required, it is informed detection; if not, it is blind detection. The process of detecting watermark has not been explained in Feng et al. (Jessica and Potkonjak 2003) or Koushanfar et al. (Koushanfar and Potkonjak 2007). Both of them only explain the process of embedding the watermark. Here, we adopt the approach of Cox et al. (Cox, Kilian, and Leighton 1997) for verifying if the watermark is present. A parallel can be drawn between the approaches of these authors and the spread-spectrum communication technology, as the watermark spreads over frequency components that are visually important. Let x be the error of the optimal solution with the message sensed data, x' the error of the optimal solution with the message sensed data and the watermark constraints, and x'' the error of the optimal solution with

the message sensed data and the watermark constraint attacks. For detecting the watermark, a correlation value or similarity measure is used in most of these methods. Here, for verifying whether the watermark signal is present, the difference measure between the normalized difference error of the optimal solution with the message sensed data and the watermark constraints, and the optimal solution with the message sensed data, is $(c = x' - x)$. The similarity measure between the normalized difference error of the optimal solution with the message sensed data and the watermark constraint attacks, and the optimal solution with the message sensed data, is $(c' = x' - x)$. The similarity measure is obtained from the normalized correlation coefficient

$$SM(C', X') = \frac{C' X'}{\sqrt{X' X'}}$$

The pseudo-code for detecting the watermark signal is shown in Pseudo Code 5.10 and the process of watermark detection in Figure 5.10.

Pseudo Code 5.8 The process of detecting watermark signal

Input

$x = [\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3]$	the error of the optimal solution with message sensed data
$x' = [\epsilon'_t, \epsilon'_{DA}, \epsilon'_{DB}, \epsilon'_{DC}, \delta'_1, \delta'_2, \delta'_3]$	the error of the optimal solution with the message sensed data and watermark constraints
$x'' = [\epsilon''_t, \epsilon''_{DA}, \epsilon''_{DB}, \epsilon''_{DC}, \delta''_1, \delta''_2, \delta''_3]$	the error of the optimal solution with the message sensed data and watermark constraints attacks
$Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$	Coefficients of objective function

Output

Detect whether watermark signal 28 bits (W) is robust or not

Step 1: Getting all input data

$$x = [\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3], x' = [\epsilon'_t, \epsilon'_{DA}, \epsilon'_{DB}, \epsilon'_{DC}, \delta'_1, \delta'_2, \delta'_3],$$

$$x'' = [\epsilon''_t, \epsilon''_{DA}, \epsilon''_{DB}, \epsilon''_{DC}, \delta''_1, \delta''_2, \delta''_3] \text{ and } Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$$

Step 2: Applying pseudo code 5.7 to get a watermark signal of 28 bits

Convert the message sensed data $Msd_1, Msd_2, Msd_3, Msd_4, Msd_5, Msd_6, Msd_7$ into 4 bits respectively, and then merge all of these to get a watermark signal of 28 bits.

Step 3: Computing the difference error of the optimal solution with the message sensed data and the watermark constraints, computing threshold, and computing similarity.

Compute $c = x' - x$ and $c' = x'' - x$, $threshold = \frac{cx'}{\sqrt{x'x'}}$ and $similarity = \frac{cx''}{\sqrt{x''x''}}$.

Step 4: Deciding whether the watermark signal of 28 bits is robust

If $threshold \geq similarity$, go to the watermark signal of 28 bits is robust

Step 5: Deciding whether the watermark signal of 28 bits is not robust

If $threshold < similarity$, go to the watermark signal of 28 bits is not robust

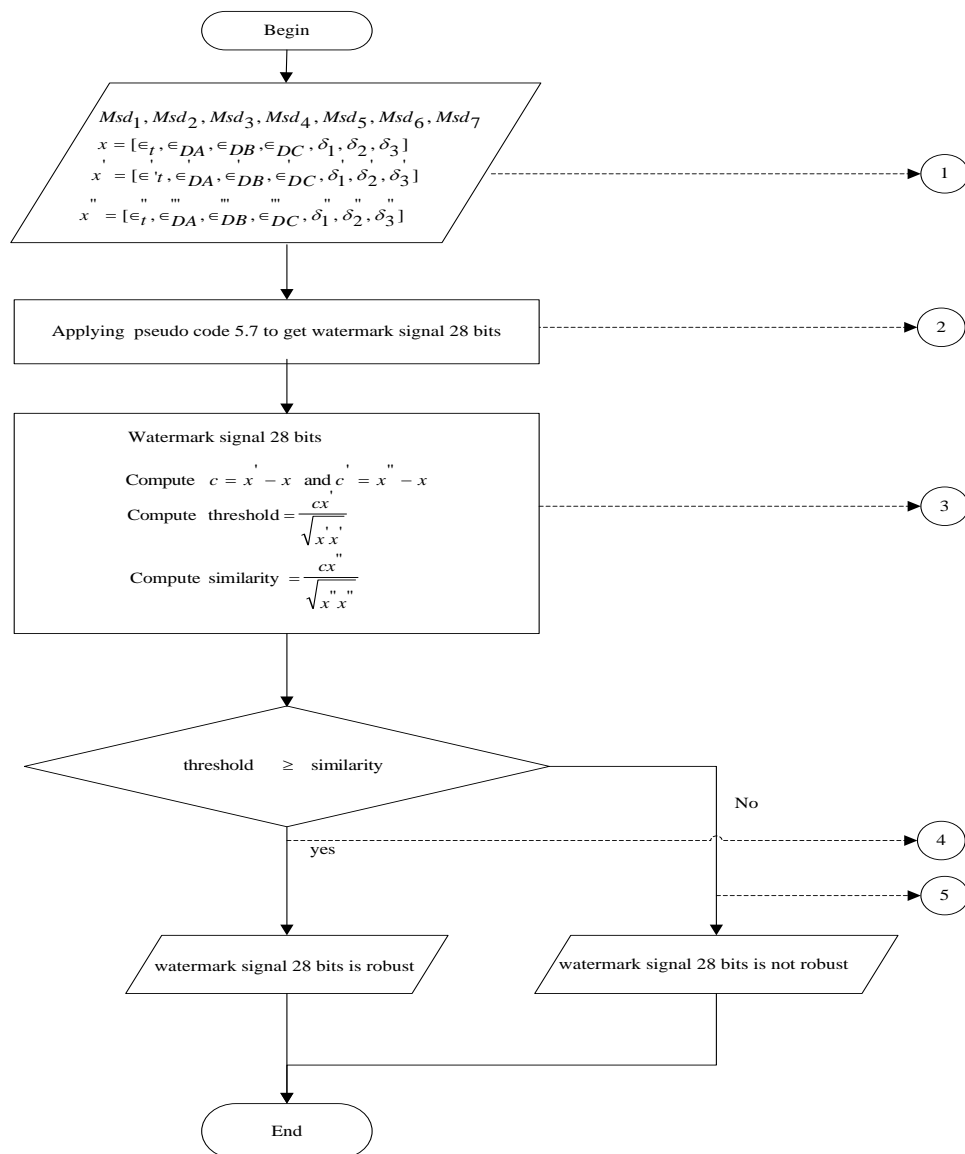


Figure 5.10 Flowchart for the process of detection of watermark signal

5.4 Implementation of the Prototype

With a view to testing the applicability of the proposed algorithm, this section describes the experimental set-up used for an extensive testing of the copyright data protection model, based on the LKR watermarking technique. This experimental set-up uses MATLAB and TOMLAB. MATLAB includes TOMLAB within it as a broad spectrum development environment for conducting researches on, and finding practical solutions of, the optimization problems. It got developed when a need came to be felt for an advanced, robust and reliable tool for developing algorithms and software, to provide solutions to various problems relating to applied optimization. This represents the fourth stage of the conceptual process described in Section 4.5.4, where the theoretical foundation was implemented as a prototype. Here, the network set-up for copyright protection, cover medium set-up, watermark generation, embedding the message sensed data, and embedding the watermark constraints and the message sensed data, have been modelled through the process of extracting watermark signal using MATLAB and TOMLAB. Some attacks to this model copyright protection scheme that attackers often use have also been modelled, such as modifying different watermark constraints using the LKR watermarking technique. The overall process of copyright protection through watermark has been modelled using MATLAB files. These are:

menuLKR.m

SetUPnetworksCopyrightWSNs.m

GenerateCoverMedium.m

Watermarkgeneration.m

EmbeddingMessageD.m

EmbeddingMsdWConstraints.m

Extract.m

Detecting.m

The operations are governed by *menuLKR.m*, which initializes all the required variables, such as the path to the network set-up, the cover medium set-up, and external applications, and passes the

embedded and the extracted parameters to the *Extract.m* file to be processed. Also, attacks are performed on the *menuLKR.m* as elaborated in Section 5.4.5. For verifying whether the watermark signal is present, the similarity found between the normalized difference error from the optimal solution and the watermarked solution X' , and the solution obtained without watermark X , is applied and modeled, using the file *detecting.m*.

5.4.1 Source Code : Network Set-Up Generation

The process of network setting is modelled using the file *SetUPnetworksCopyrightWSNs.m*. The call to *SetUPnetworksCopyrightWSNs.m* from *menuLKR.m* passes to the location of the network setting. The pseudo code for network setting process has been explained in greater detail in Section 5.5.1.

5.4.2 Source Code: Cover Medium Generation

The process of cover medium generation is modelled using the file *GenerateCoverMedium.m*. The call to *GenerateCoverMedium.m* from *menuLFSKR.m* passes to the location of the cover medium, and the parameters for the cover medium to perform the embedding operation are determined. The pseudo code for the cover medium generation has been explained in Section 5.3.1. Figure 5.11 shows the screenshot of the cover medium generation process using the MATLAB Code

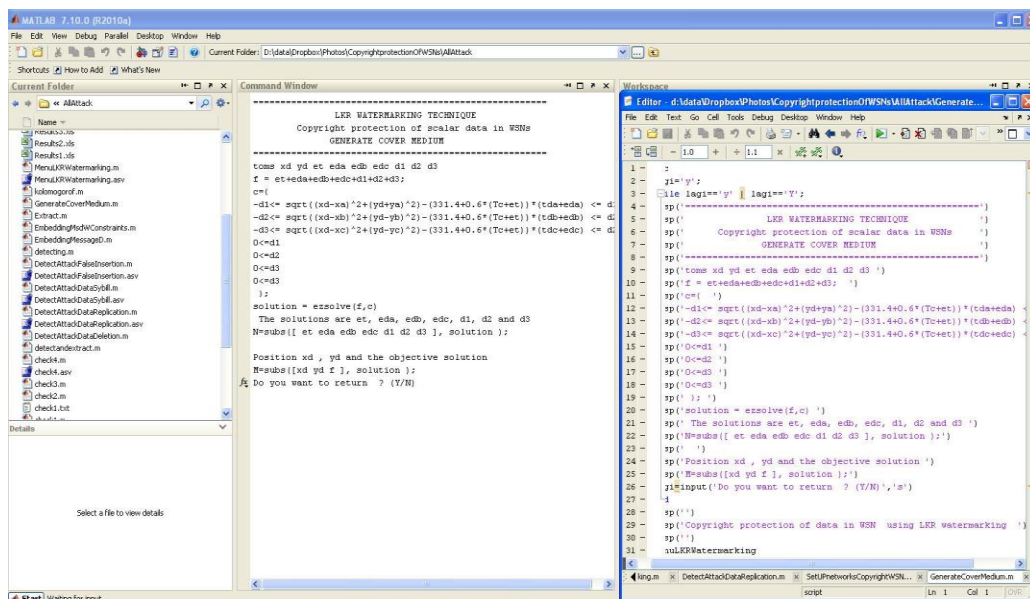


Figure 5.11 Screenshot of the cover medium generation process using MATLAB code

5.4.3 Source Code : Watermark Generation

The process of watermark generation consists of four steps, i.e., converting the sensitive data modelled through the MATLAB function *de2bi.m*, generating the watermark signal modelled using the file *LFSR.m*, producing watermark constraints modelled through the file *KolmogorovRule.m*, and generating the message sensed data modelled using the file *MessageSenseData.m*. The pseudo code for the watermark generation process has been explained in Section 5.3.2. Figure 5.12 shows the screenshot of the process of generating the watermark signal and the watermark constraints using the MATLAB code.

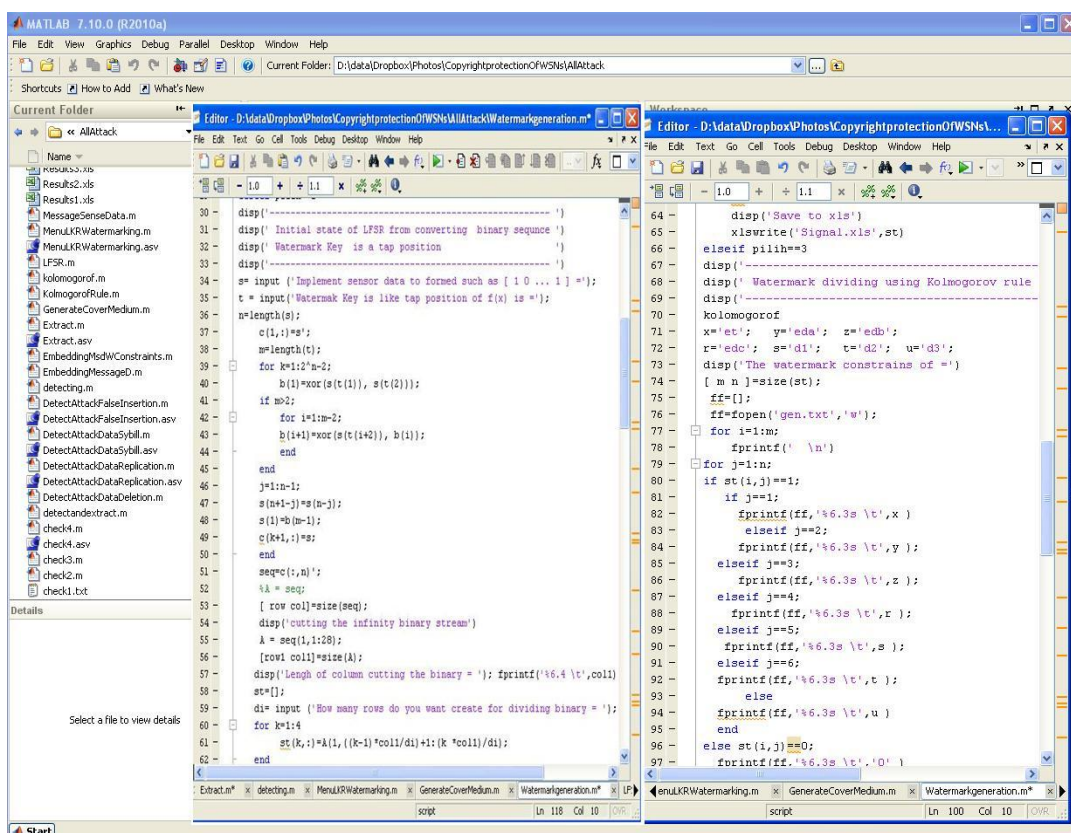
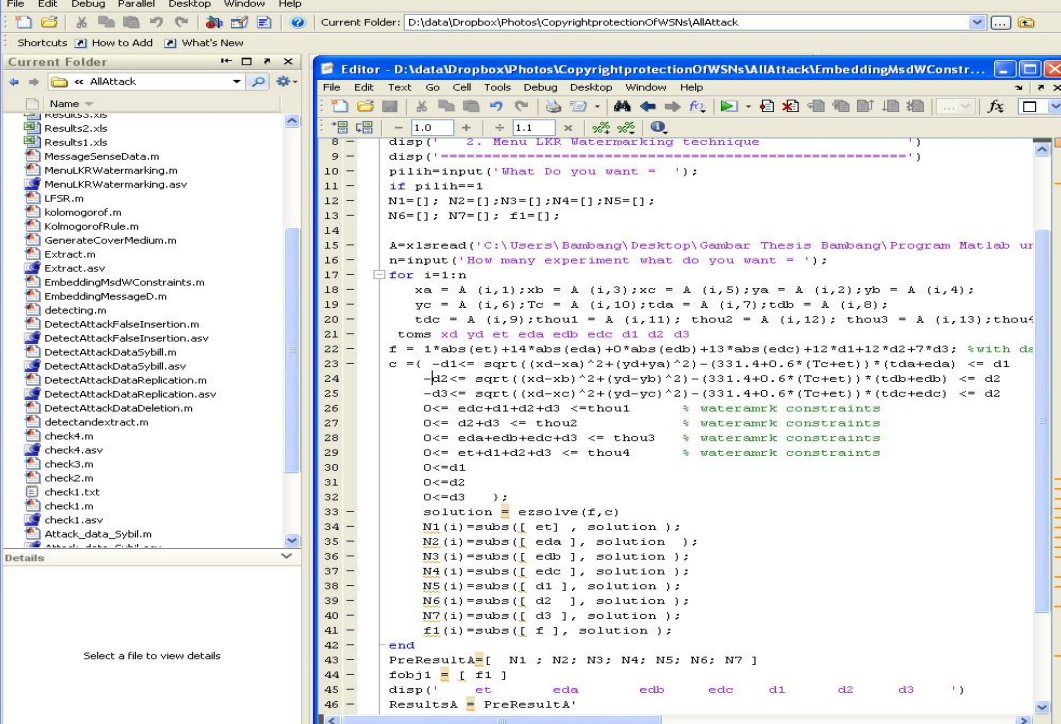


Figure 5.12 Screenshot of the process of generating watermark signal and watermark constraints using MATLAB code

5.4.4 Source Code : Embedding Watermark Signal

The process of embedding a watermark signal consists of two steps - embedding watermark constraints and embedding message sensed data. The process of embedding watermark constraints is modelled using the file *EmbeddingMsdWConstraints.m*, while the process of embedding message sensed data is modelled using the file *EmbeddingMessageD.m*.

The call to both *EmbeddingMsdWConstraints.m* and *EmbeddingMessageD.m* from *menuLKR.m* passes to the location of the cover medium, and the parameters to perform the embedding operation are determined. The pseudo code for embedding watermark constraints and message sensed data have been elaborated in Section 5.3.3. Figure 5.13 shows the screenshot of the process of embedding the watermark signal using the MATLAB code.



```

8 disp(' 2. Menu LKR Watermarking technique')
9 disp('-----')
10 pilih=input('What Do you want = ');
11 if pilih==1
12 N1=[]; N2=[];N3=[];N4=[];N5=[];
13 N6=[]; N7=[]; f1=[];
14
15 A=x1read('C:\Users\Bambang\Desktop\Gambar Thesis Bambang\Program Matlab ur
16 n=input('How many experiment what do you want = ');
17 for i=1:n
18 xa = A (1,1);xb = A (1,3);xc = A (1,5);ya = A (1,2);yb = A (1,4);
19 yc = A (1,6);Tc = A (1,10);tda = A (1,7);tdb = A (1,8);
20 tdc = A (1,9);thou1 = A (1,11); thou2 = A (1,12); thou3 = A (1,13);thou4
21 toms xd yd et eda edb edc d1 d2 d3
22 z = 1*abs(et)+14*abs(eda)+40*abs(edb)+13*abs(edc)+12*d1+12*d2+7*d3; %with ds
23 c = [-d1<= sqrt((xd-xa)^2+(yd-ya)^2)-(331.4+0.6*(Tc+et))* (tda+eda) <= d1
24 -d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))* (tdb+edb) <= d2
25 -d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))* (tdc+edc) <= d2
26 0<= edc+d1+d2+d3 <=thou1 % watermark constraints
27 0<= d2+d3 <= thou2 % watermark constraints
28 0<= eda+edb+edc+d3 <= thou3 % watermark constraints
29 0<= et+d1+d2+d3 <= thou4 % watermark constraints
30 0<=d1
31 0<=d2
32 0<=d3
33 solution = ezsolve(f,c)
34 N1(i)=subs([ et ], solution );
35 N2(i)=subs([ eda ], solution );
36 N3(i)=subs([ edb ], solution );
37 N4(i)=subs([ edc ], solution );
38 N5(i)=subs([ d1 ], solution );
39 N6(i)=subs([ d2 ], solution );
40 N7(i)=subs([ d3 ], solution );
41 f1(i)=subs([ f ], solution );
42 end
43 PreResultA=[ N1 ; N2; N3; N4; N5; N6; N7 ]
44 fobj1 = [ f1 ]
45 disp(' et eda edb edc d1 d2 d3 ')
46 ResultsA = PreResultA'

```

Figure 5.13 Screenshot of the process of embedding watermark signal using MATLAB code

5.4.5 Source Code: Extracting Watermark

The process of extracting watermark signal is modelled using the file *Extract.m*. The pseudo code for the process of extracting the watermark signal has been elaborated in Section 5.3.4. Figure 5.14 shows the screenshot of extracting the watermark signal using the MATLAB code.

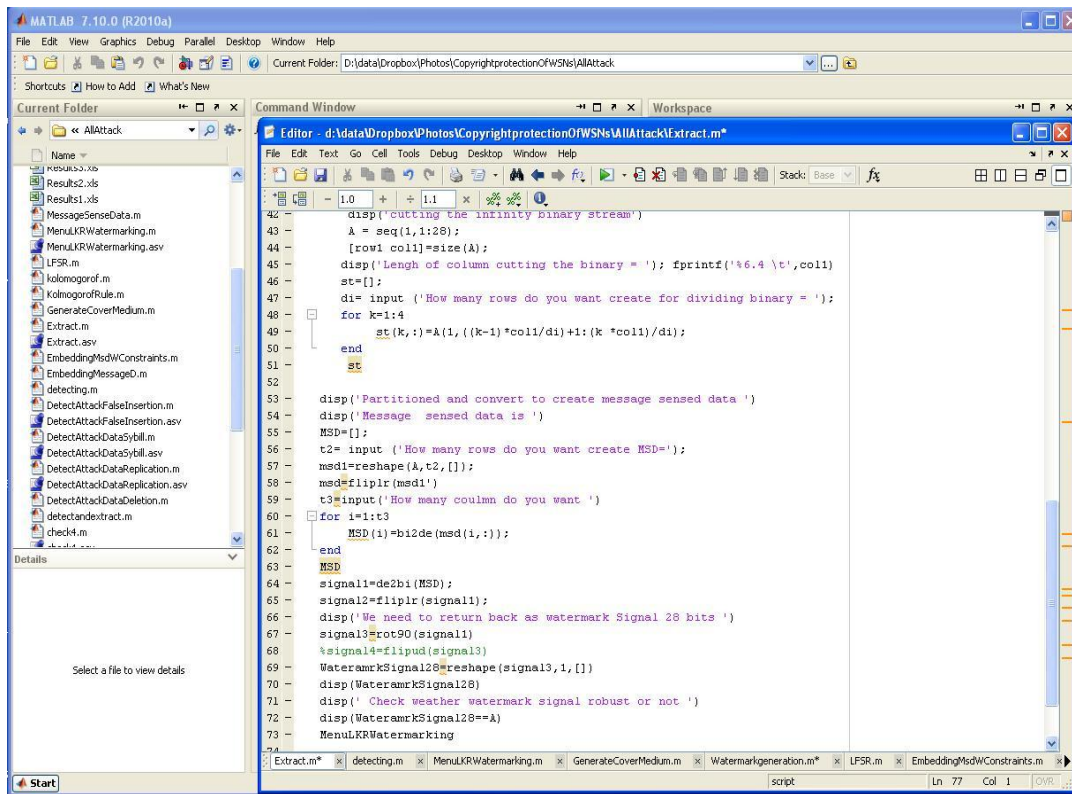


Figure 5.14 Screenshot of the process of extracting watermark signal using MATLAB code

5.4.6 Source Code: Attacks

We assume that the attacker can estimate the watermark constraints, which, therefore, should be modified and changed. The corresponding watermark constraint attacks carried out in our experiment are:

1. Inserting a number of different watermark constraints, generated using the LFSR and Kolmogorov rule, hoping to find new results for the error of cover medium that will be mapped into the existing solution. This process of insertion of a number of different attacks is modelled using the file *Attack_data_False_Insertion.m*.
2. Deleting a number of watermark constraints hoping to find new results for the error of cover medium that will be mapped into the existing solution. This process of deletion of a number of watermark attacks is modelled using the file *Attack_data_Deletion.m*.
3. Replicating different watermark constraints, generated using the LFSR and Kolmogorov rule, hoping to find new results for the error of cover medium that will be mapped into the existing solution. This process of replication of watermark constraints is modelled using the file *Attack_data_replication.m*.

4. Presenting more than one attacker identity within the network by creating more than one watermark constraint, hoping to find new results for the error of cover medium that will be mapped into the existing solution. This process of presenting more than one identity attacker is modelled using the file *Attack_data_Sybil.m*.

All the implementation details of *Attack_data_False_Insertion.m*, *Attack_data_Deletion.m*, *Attack_data_replication.m* and *Attack_data_Sybil.m* have been provided in Section 5.5.3.

5.5 Experimental Setting

This section describes the experimental set-up used for an extensive testing of our copyright data protection model. In it, 75 nodes were placed randomly within a 500 square meter area, as shown in Figure 5.10.

5.5.1 Network Set -Up

This section explains the process of setting up the wireless sensor network using MATLAB simulation. The network consists of 75 sensor nodes placed randomly within a 500 square meter area. In all, 32 positions of (x_A, y_A) , (x_B, y_B) , and (x_C, y_C) were generated, using the random positions of 75 sensor nodes, generating 32 of τ_c using gauss distribution on interval [0,1], 32 of t_{DA} , t_{DB} , and t_{DC} using uniform distribution on interval [0,1], and 32 of τ_1, τ_2, τ_3 and τ_4 using gauss distribution on interval [0,1], in order to avoid the possibility of these values hampering a feasible cover medium solution.

The pseudo-code for setting up the network is shown in Pseudo Code 5.9.

Pseudo Code 5.9 Network Set-up for LKR Watermarking Technique

Input

N	Number of Sensor Nodes
L	Placing area within a unit square of sensor nodes
R	Maximum range of two sensor nodes communicate

Output

$(x_i, y_i), i=1, \dots, 75$	Coordinate Position of two-dimensional sensor nodes
------------------------------	---

T_c	Temperature of the propagation media
$t_{DA}, t_{DB},$ and t_{DC}	Generate time transmission between two sensor nodes
δ_1, δ_2 and δ_3	Errors between the Euclidean distances measured the feasibility of the solution space of the optimization
$\tau_1, \tau_2, \tau_3, \tau_4$	The value of the feasibility of the solution space of the optimization

Step 1: Setting up all data

$N=75, L = 500$ and $R=50$

Set N number of sensor nodes randomly within an L square meter area and set the maximum range of two sensor nodes communicate as R communicating as R .

Step 2: Positioning and plotting coordinates

Generate coordinates of 75 sensor nodes' positions of x, y and plot these coordinates

Step 3: Plotting a link between two sensor nodes using Euclidean Theorem

Compute distance between two sensor nodes using Euclidean Theorem. If distance is less than R then there is a link between the two sensor nodes, else there is no link between the two sensor nodes

Step 4: Generating parameters of time transmission, temperature, errors between two Euclidean distances and the value feasibility of the solution space

Generate time transmission between two sensor nodes based on uniform distribution (0,1), generate temperature of the propagation media based on random between (1,75), generate errors between the Euclidean distance measures based on uniform distribution (0,1), and the feasibility of the solution space of the optimization based on uniform distribution (0,1)

Step 5: Printing coordinate position, temperature, time transmission, error between two distance measures, and the value of the visibility

Print coordinate position $(x_i, y_i), i = 1, 2, 3, \dots, 75$, temperature T_c , time transmission t_{DA}, t_{DB}, t_{DC} , and the value of the visibility $\tau_1, \tau_2, \tau_3,$ and τ_4

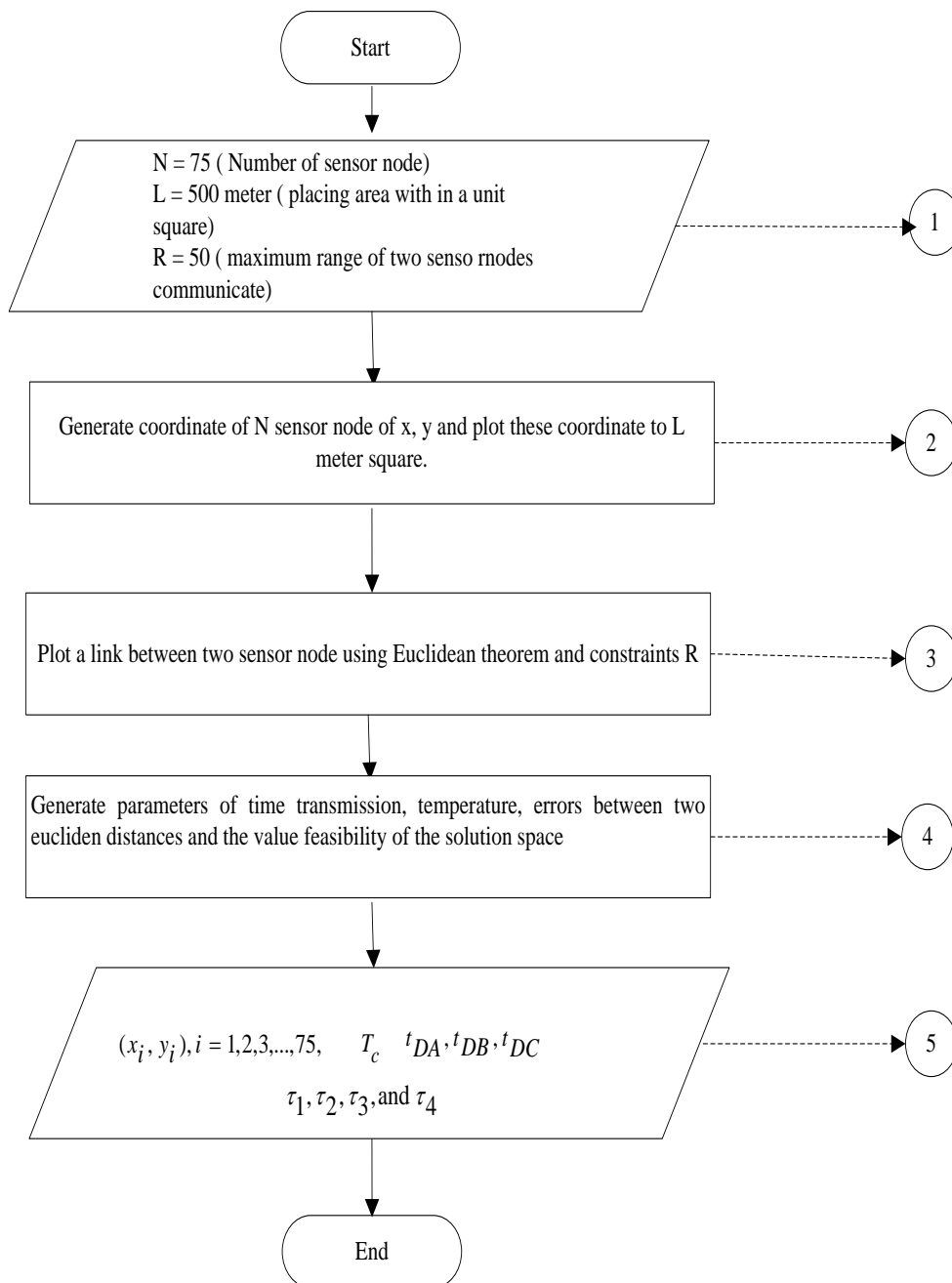


Figure 5.15 Flowchart for setting up LKR watermarking technique

The network set-up for the LKR watermarking technique was implemented through pseudo code 5.6 using MATLAB. Figure 5.15 shows the screenshot of the network setting of the LKR Watermarking Matlab Code, and Figure 5.16 shows the 75 nodes randomly deployed within a 500 square meter area.

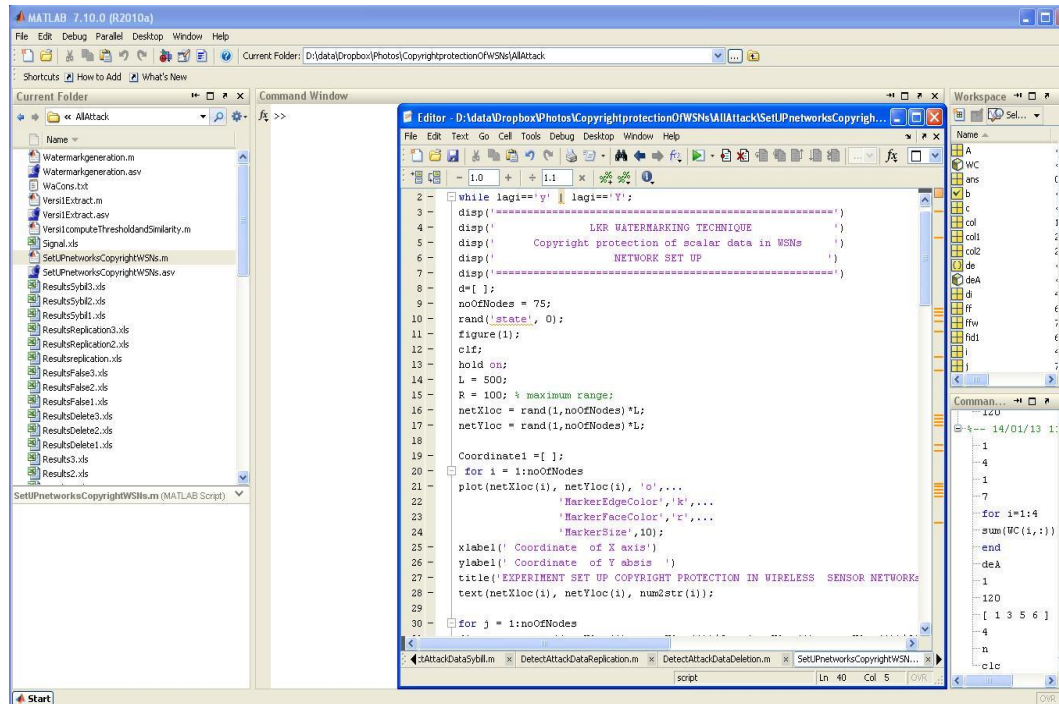


Figure 5.16 Screenshot of the network setting for LKR Watermarking Technique using MATLAB Code

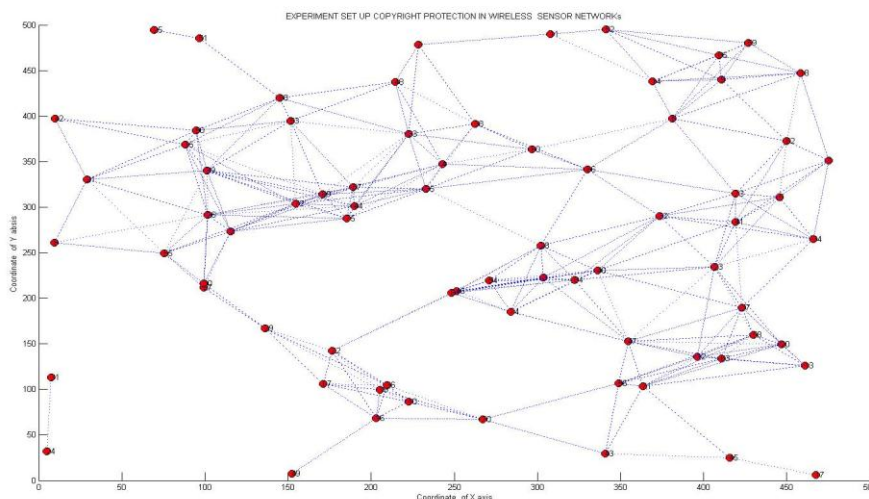


Figure 5.17 The 75 nodes randomly deployed within a 500 meter square area

Based on Figure 5.16, the coordinates of 75 nodes can be listed, as given in Table 5.2

Table 5.2 Coordinate positions of 75 nodes

No	X	Y	No	X	Y	No	X	Y
1	475.0646	351.37	26	202.7652	970.8449	51	193.4312	451.4248
2	115.5693	273.2856	27	198.7217	990.0826	52	682.2232	43.89533
3	303.4213	222.4401	28	603.7925	788.8617	53	302.7644	27.18512
4	242.9912	347.2836	29	272.1879	438.6585	54	541.6739	312.685
5	445.6495	310.6551	30	101.3826	291.3958	55	96.71558	485.4225
6	381.0484	397.4105	31	99.36087	211.7481	56	341.1116	495.0413
7	228.2338	478.4217	32	301.8962	257.7559	57	151.3822	394.4308
8	9.251822	261.2952	33	136.094	166.9757	58	270.8369	219.3293
9	410.7036	440.0711	34	99.40713	216.4533	59	75.43649	249.1557
10	222.3517	86.47807	35	7.636964	112.9749	60	348.9492	106.9817
11	307.7162	489.8734	36	373.3928	289.9034	61	189.1865	321.7461
12	395.9685	135.7236	37	222.5482	380.1825	62	430.0058	160.0178
13	460.9065	126.1647	38	465.9073	264.9116	63	426.8276	480.0493
14	369.1036	437.8709	39	232.9972	320.2632	64	296.7815	363.3159
15	88.13307	368.653	40	209.3247	104.5347	65	248.2762	205.9766
16	202.8531	68.25937	41	423.1107	189.9092	66	449.8846	372.2829
17	467.7348	5.878344	42	262.5762	391.6643	67	410.8146	133.9736
18	458.4522	446.949	43	101.3237	340.4229	68	322.4552	219.9622
19	205.1351	99.56903	44	336.0687	230.5476	69	408.9872	466.6901
20	446.8248	149.3615	45	419.0592	283.9144	70	330.1138	341.6662
21	28.94565	330.7213	46	9.819757	397.1053	71	170.9853	106.2799
22	176.4341	142.2043	47	340.6386	29.5913	72	144.8629	419.6191
23	406.5832	234.6121	48	189.7405	301.4345	73	170.5968	314.3923
24	4.93065	32.39056	49	415.898	25.1344	74	267.0395	66.88637
25	69.44544	494.1675	50	251.4064	207.6874	75	363.5566	103.5664

Now that we have obtained a network setting and a list of the coordinates of 75 random points within a 500 square, we will generate a cover medium.

The experimental setting of the cover medium uses a cover medium algorithm based on Pseudo Code 5.2, which was explained in Section 2.2.1. We obtained the cover medium NLSP as follows:

$$\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3$$

Constraints

$$\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3$$

$$0 \leq \delta_1$$

$$0 \leq \delta_2$$

$$0 \leq \delta_3$$

We next obtain all the data input to be used for preparing the cover medium. In all, we 32 data inputs by:

1. generating 32 $(x_A, y_A), (x_B, y_B)$ and (x_C, y_C) using the random positions of 75 sensor nodes within a 500 square meter area,
2. generating 32 of t_{DA}, t_{DB} and t_{DC} using a uniform distribution on interval [0,1],
3. generating 32 of τ_c using gauss distribution on interval [0,1],
4. generating 32 of τ_1, τ_2, τ_3 and τ_4 using gauss distribution on interval [0,1], in order to avoid these values hampering a feasible cover medium solution.

Table 5.3 shows 32 experiments of data input for copyright protection in WSNs.

Table 5.3 32 experiments of the positions of $(X_A, Y_A), (X_B, Y_B), (X_C, Y_C)$ and 32 experiments of time measurement using Gaussian distribution [0,1], 32 experiments of temperature randomly between [0,75] and the values of τ_1, τ_2, τ_3 , and τ_4 generated using normal distribution [0,1].

No	Position of three sensor nodes								The exact time				temperature		The feasibility of value			
	Xa	Ya	Xb	Yb	Xc	Yc	Tda	Tdb	Tdc	Tde	temp	theta1	theta2	theta3	theta4			
1	475.0646	351.37	242.9912	347.2836	228.2338	478.4217	0.771625	0.106793	0.09282	57.6720658	0.175196	0.885999	0.972824	0.792262				
2	115.5693	273.2856	445.6495	310.6551	9.251822	261.2952	0.171564	0.464945	0.345087	17.70893722	0.145088	0.720987	0.929214	0.267388				
3	305.4213	222.4401	381.0484	397.4105	410.7036	440.0711	0.375385	0.066181	0.742588	65.73479994	0.085979	0.172273	0.686	0.681904				
4	242.9912	347.2836	228.2338	478.4217	222.3517	86.47807	0.883441	0.328065	0.774966	67.90485822	0.226608	0.532973	0.08596	0.707953				
5	445.6495	310.6551	9.251822	261.2952	307.7162	489.8734	0.527153	0.593126	0.989568	74.29563351	0.143694	0.12612	0.364157	0.550297				
6	381.0484	397.4105	410.7036	440.0711	395.9685	135.7236	0.908981	0.236647	0.3688	48.32886399	0.132453	0.66292	0.631822	0.523544				
7	228.2338	478.4217	222.3517	86.47807	460.9065	126.1647	0.55177	0.566397	0.400609	52.50678122	0.1752	0.37333	0.857519	0.78676				
8	9.251822	261.2952	307.7162	489.8734	369.1036	437.8709	0.645078	0.769815	0.558389	15.58392425	0.502445	0.696362	0.656257	0.164214				
9	410.7036	440.0711	395.9685	135.7236	88.13307	368.653	0.437583	0.452116	0.639911	61.91649872	0.139248	0.529794	0.133427	0.393352				
10	222.3517	86.47807	460.9065	126.1647	202.8531	68.25937	0.762819	0.279458	0.885052	11.30541018	0.418395	0.894145	0.812015	0.377486				
11	307.7162	489.8734	369.1036	437.8709	467.7348	5.878344	0.807509	0.788252	0.33704	13.81494052	0.258256	0.043369	0.21449	0.462476				
12	395.9685	135.7236	88.13307	368.653	458.4522	446.949	0.267034	0.185511	0.183031	16.99805197	0.929905	0.094724	0.090759	0.825058				
13	460.9065	126.1647	202.8531	68.25937	205.1331	99.56903	0.899176	0.398851	0.6772	22.07859828	0.211483	0.494799	0.675361	0.558794				
14	369.1036	437.8709	467.7348	5.878344	446.8248	149.3615	0.929729	0.549408	0.026827	21.71466257	0.162104	0.410317	0.528332	0.372826				
15	88.13307	368.653	458.4522	446.949	28.94565	330.7213	0.58159	0.030004	0.884546	72.17560894	0.240593	0.554513	0.260658	0.179209				
16	202.8531	68.25937	205.1331	99.56903	176.4341	142.2043	0.667822	0.642328	0.205282	35.81519147	0.454819	0.41874	0.45097	0.276001				
17	467.7348	5.878344	446.8248	149.3615	406.5832	234.6121	0.257769	0.994134	0.56588	74.76066258	0.937842	0.388954	0.539642	0.657869				
18	458.4522	446.949	28.94565	330.7213	4.93065	32.39056	0.764466	0.766301	0.937439	10.93661768	0.791832	0.121336	0.764885	0.979257				
19	205.1331	99.56903	176.4341	142.2043	69.44544	494.1675	0.025326	0.282627	0.238757	5.807154105	0.675217	0.823526	0.411555	0.017454				
20	446.8248	149.3615	406.5832	234.6121	101.3826	291.3958	0.160745	0.984648	0.648517	69.67010293	0.051743	0.283906	0.503754	0.846913				
21	28.94565	330.7213	4.93065	32.39056	99.36087	211.7481	0.124422	0.801277	0.778655	60.01689411	0.853475	0.404699	0.098869	0.541064				
22	176.4341	142.2043	69.44544	494.1675	301.8962	257.7559	0.422712	0.909622	0.727889	15.90433244	0.414134	0.483438	0.469516	0.420039				
23	406.5832	234.6121	101.3826	291.3958	136.094	166.9757	0.551549	0.333024	0.068103	33.6860622	0.292539	0.197008	0.131023	0.316832				
24	4.93065	32.39056	99.36087	211.7481	99.40713	216.4533	0.457755	0.425016	0.646568	3.424007987	0.102947	0.274277	0.001126	0.680634				
25	69.44544	494.1675	301.8962	257.7559	7.636964	112.9749	0.093361	0.237716	0.64726	54.7832739	0.249484	0.628337	0.209496	0.642622				
26	101.3826	291.3958	136.094	166.9757	373.3928	289.9034	0.987543	0.994171	0.080329	69.03854238	0.845984	0.08568	0.700388	0.669692				
27	99.36087	211.7481	99.40713	216.4533	222.5482	380.1825	0.26841	0.295383	0.132956	6.371329013	0.246714	0.495262	0.679907	0.893374				
28	301.8962	257.7559	7.636964	112.9749	465.9073	264.9116	0.033219	0.248909	0.315704	52.24177398	0.732167	0.164212	0.26915	0.892367				
29	136.094	166.9757	373.3928	289.9034	232.9972	320.2632	0.688727	0.514439	0.483494	61.12785874	0.785673	0.291884	0.403719	0.458071				
30	99.40713	216.4533	222.5482	380.1825	209.3247	104.5347	0.565756	0.740018	0.406715	35.8146301	0.007369	0.13551	0.590197	0.914413				
31	7.636964	112.9749	465.9073	264.9116	423.1107	189.9092	0.932561	0.263116	0.432125	31.5470358	0.858687	0.379987	0.095788	0.14322				
32	373.3928	289.9034	232.9972	320.2632	262.5762	391.6643	0.706454	0.44255	0.267831	55.94139156	0.246189	0.790054	0.680811	0.139942				

In the next experimental setting, we obtain sensitive scalar data. It is this scalar data which is to be protected using the LKR watermarking technique. The scalar data have been captured by the sensor nodes and number 120. We use the code of watermark generation as elaborated in Section 5.4.3. Figure 5.18 shows the screenshot of the process of producing the watermark signal (W), watermark constraints (WC) and Message Sensed Data (MSD).

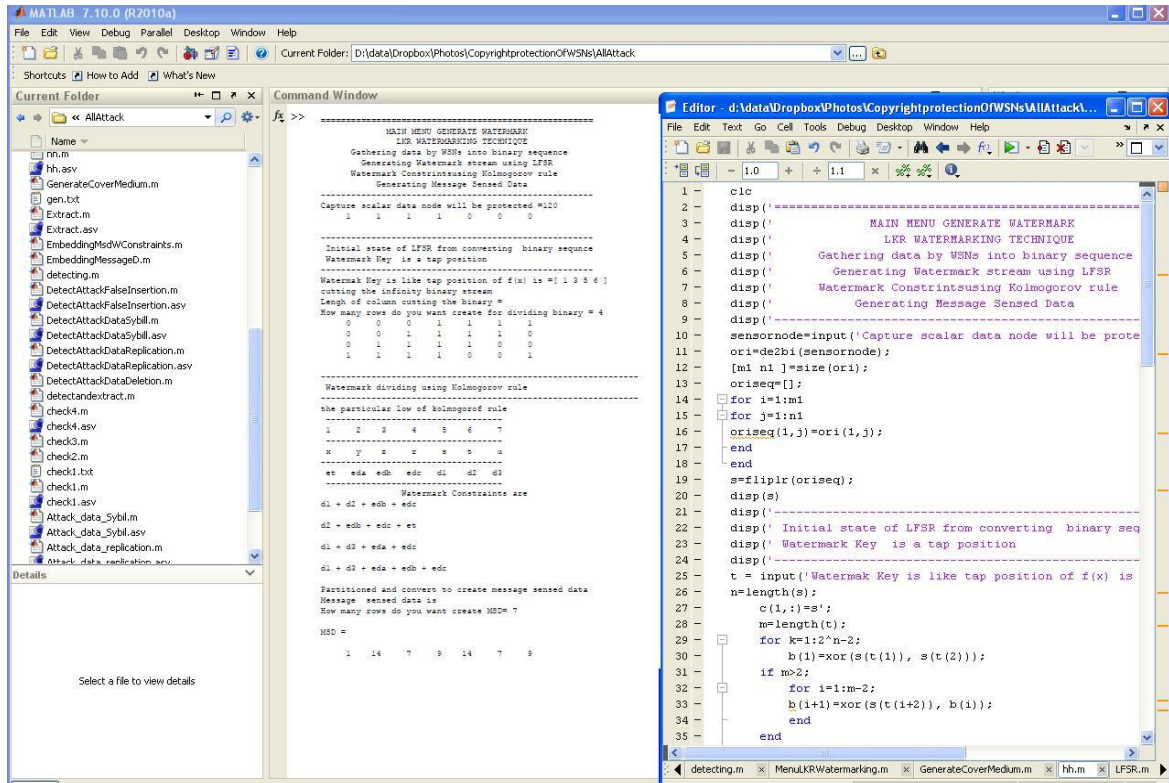


Figure 5.18 Screenshot of generating watermark signal MATLAB Code

```

=====
MAIN MENU GENERATE WATERMARK
LKR WATERMARKING TECHNIQUE
Gathering data by WSNs into binary sequence
Generating Watermark stream using LFSR
Watermark Constraints using Kolmogorov rule
Generating Message Sensed Data

-----
Which one do you want = 1
-----

Gathering data by WSNs into binary sequence
Converting the data sensory decimal to binary sequence
-----

Capture scalar data node =120
    
```

```
s =   1   1   1   1   0   0   0
```

```
-----
Initial state of LFSR from converting binary sequence
Watermark Key is a tap position
```

```
-----
Watermak Key is like tap position of f(x) is =[ 1 3 5 6 ]
cutting the infinity binary stream
```

```
st =
```

```
   0   1   0   1   1   0   1
   0   1   1   0   1   1   0
   0   1   1   0   1   1   0
   1   0   1   1   0   1   1
```

```
-----
Watermark dividing using Kolmogorov rule
```

```
-----
The particular low of kolmogorof rule
```

```
-----
1     2     3     4     5     6     7
-----
```

```
x     y     z     r     s     t     u
-----
```

```
et   eda  edb  edc  d1   d2   d3
-----
```

```
The watermark constrains of =
```

```
d1 + d2 + eda + edb + edc
```

```
eda + edb + et
```

```
d1 + d2 + d3
```

```
d1 + d2 + d3 + eda + edb + edc
```

```
message sensed data is
```

```
MSD =
```

```
   1   14   7   9   14   7   9
```

At the end of the experiment, we partition and convert the Message Sensed Data (MSD) into watermark signal. We use Pseudo Code 5.7 to convert the MSD into watermark signal. Figure 5.19 shows the screenshot of the processes of generating and extracting the signal using MATLAB Code.

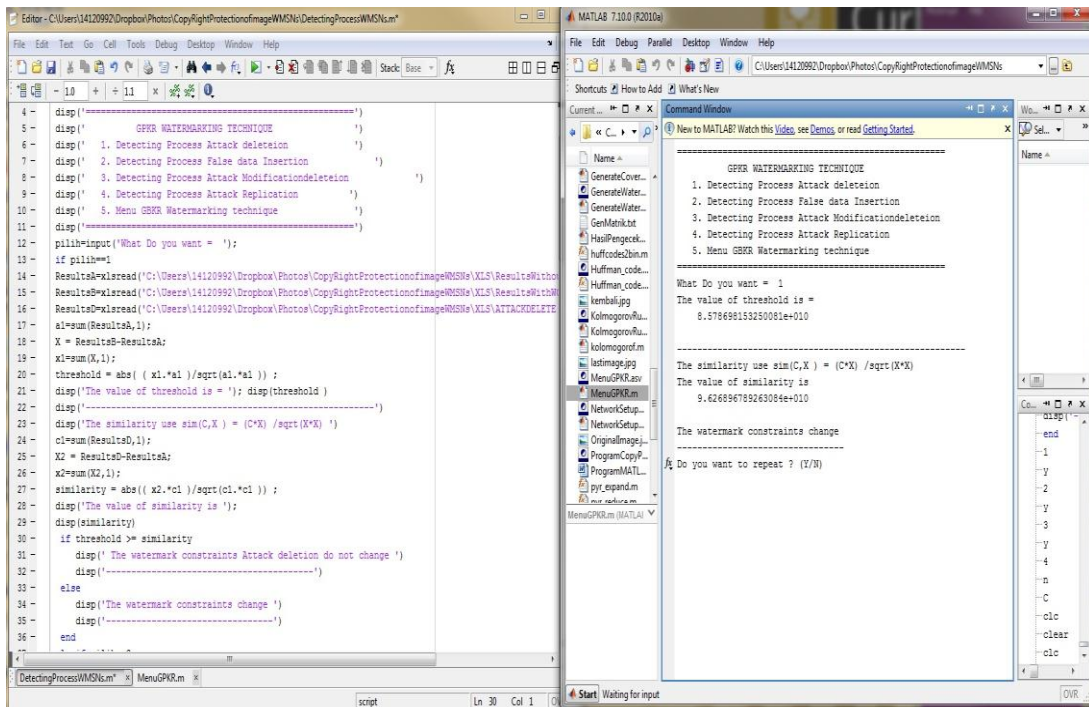


Figure 5.19 Screenshot of generating and extracting the signal using MATLAB Code

```

=====
LKR WATERMARKING TECHNIQUE
Copyright protection of scalar data in WSNs
1. Partitioned and convert to create MSD
2. Menu LKR Watermarking technique
=====

Read Message Sensed Data (MSD) is =
Message sensed data is
    1    14     7     9    14     7     9

We need to return back as watermark Signal 28 bits
0     0     0     1     1     1     1     0     0     1     1
1     1     0     0     1     1     1     1     0     0     1
1     1     1     0     0     1
    
```

5.5.2 Parameters

lists all the parameters and their associated values used for capturing the results from the algorithm.

Table 5.4 Parameter and its associated values used in the LKR watermarking technique lists all the parameters and their associated values used for capturing the results from the algorithm.

Table 5.4 Parameter and its associated values used in the LKR watermarking technique

Parameter	Description	Metric
N	Number of sensor node	Integer
(x_i, y_j) $i = j = 1, 2, \dots, n$	Position of two-dimensional sensor networks	Coordinate
T_c	Temperature of the propagation media	Degree
t_{DA}, t_{DB}, t_{DC}	Time transmission between node D to A, D to B and D to C	second
V_s	Speed acoustic signal	(m/s)
ϵ_t	Error in the measurement of the temperature	-
$\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$	Error in the measurement of the timer from D to A, D to B and D to C	-
$\delta_1, \delta_2, \delta_3$	the error in the measurement between the Euclidean measurement and the measured using time differences of optimal D to A, D to B and D to C.	-
$\tau_1, \tau_2, \tau_3, \tau_4$	Values are selected such that the feasibility of the solution space of the optimization problem is not harmed	-
Sensed data	Data sensed by a sensor node	Bit
<i>threshold</i>	Normalized correlation the results of error the cover medium with watermark constraints	-
$SM(C', X')$	Normalized correlation the results of error the cover medium with watermark constraints attack	-
Watermark signal	Result from LFSR	Bit
Message sensed data	Result from pseudo code 4	Integer
Watermark constraints	Result from pseudo code	-

5.5.3 Attacks Characterization

Because of the nature of the transmission medium that broadcasts data, WSNs are quite susceptible to security attacks. Additionally, the nodes are often deployed in unattended or even hostile environments and cannot be physically protected. An attack is said to have succeeded if the

receiver is not able to detect it. This section discusses different kinds of attacks possible on a WSN, and explains how our protection scheme resists them.

Consider the general model of a watermarking technique for copyright protection for WSNs, according to a communication formulation, with its block diagram as shown in Figure 5.1. We consider in detail the corresponding weaknesses of the LKR watermarking technique that could be used by the attacker. Figure 5.20 shows the categorization of attacks that affect communication.

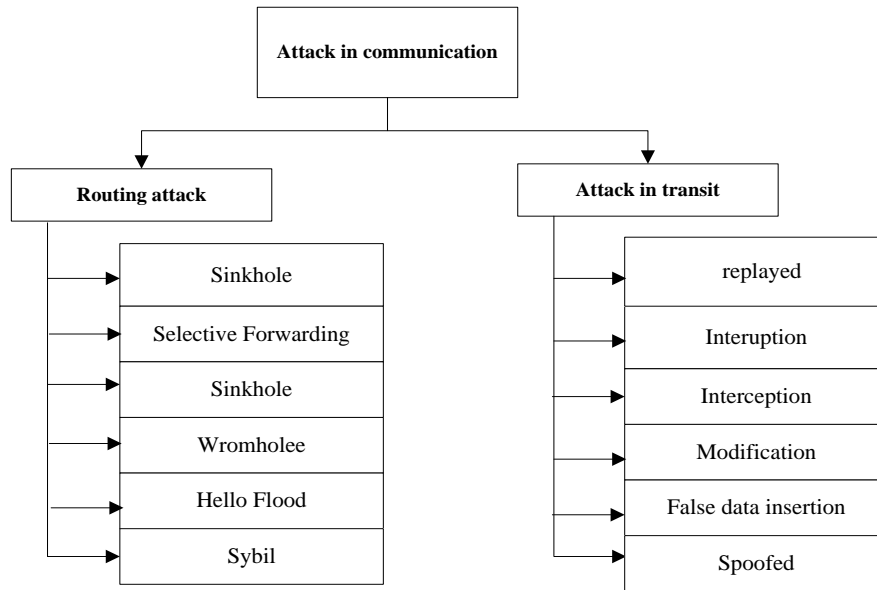


Figure 5.20 The categorization of attacks that affect communication.

Assume that the watermarks constraints are estimated by the attacker that should be change, modify, manipulation, replayed and remove. The corresponding attacks are:

5.5.3.1 False Data Insertion Attack

In false data insertion attack, an attacker compromises certain nodes and inserts false data into the network. Insertion refers to gradually of the attacker's presence. It is called true if the attacker can attack the network at the level of sensor nodes, otherwise false. The attacker fools the cover medium to make a wrong security decision by creating misleading contexts. A false data insertion attack is like cheating the network: the data set up into it in the form of constraints are false, yet convincing. The cover medium's response would have been right had the data been real, but it is not, because the data is not real. False data insertion attack can be carried out if the attacker is somehow able to compromise the existing nodes' security in order to inject false message sensed data carrying false information. Figure 5.21 shows the insertion of false watermark constraints into

the cover medium. The objective of this attack is to try a number of different watermark constraints generated by the LFSR, hoping to get new results of error of the cover medium that will be mapped into the existing solution.

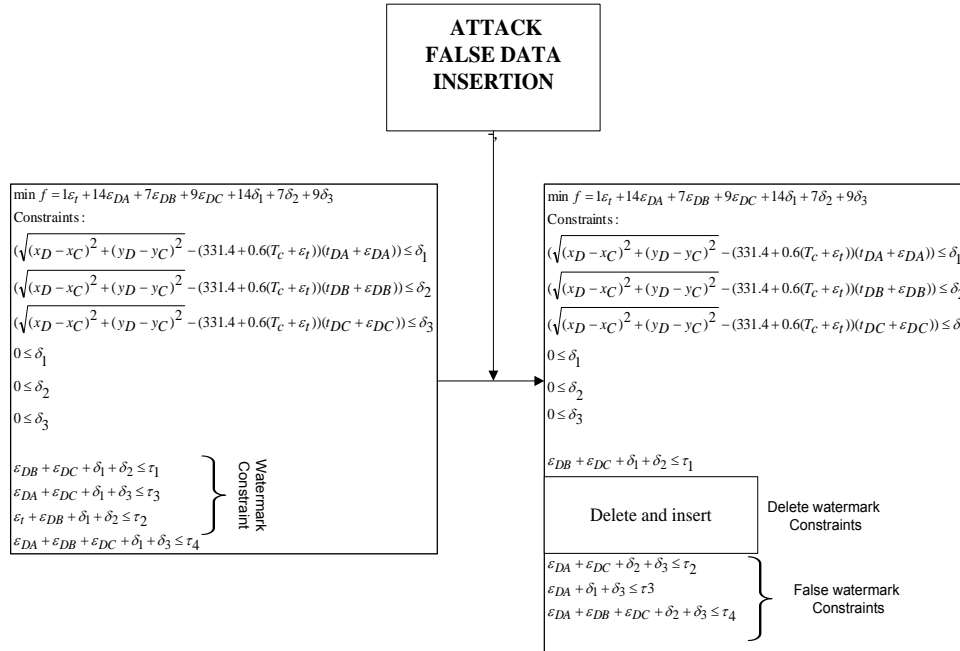


Figure 5.21 Insertion of false watermark constraints into the cover medium

5.5.3.2 Data Deletion Attack

Data deletion attack is similar to the spoofed data attack in the sense that deleting watermark constraints makes the error results of the cover medium invalid. So, the watermark signal is also invalid because it does not approximate to the results of errors of cover medium without attack. If the attacker deletes watermark constraints, the receiver will not get appropriate results of errors. To carry out such an attack, the attacker drops the individual watermark constraint readings, or some of the watermark constraints. These constraints are, thus, unable to reach the intended recipient. Figure 5.22 shows deletion of watermark constraint data from the cover medium.

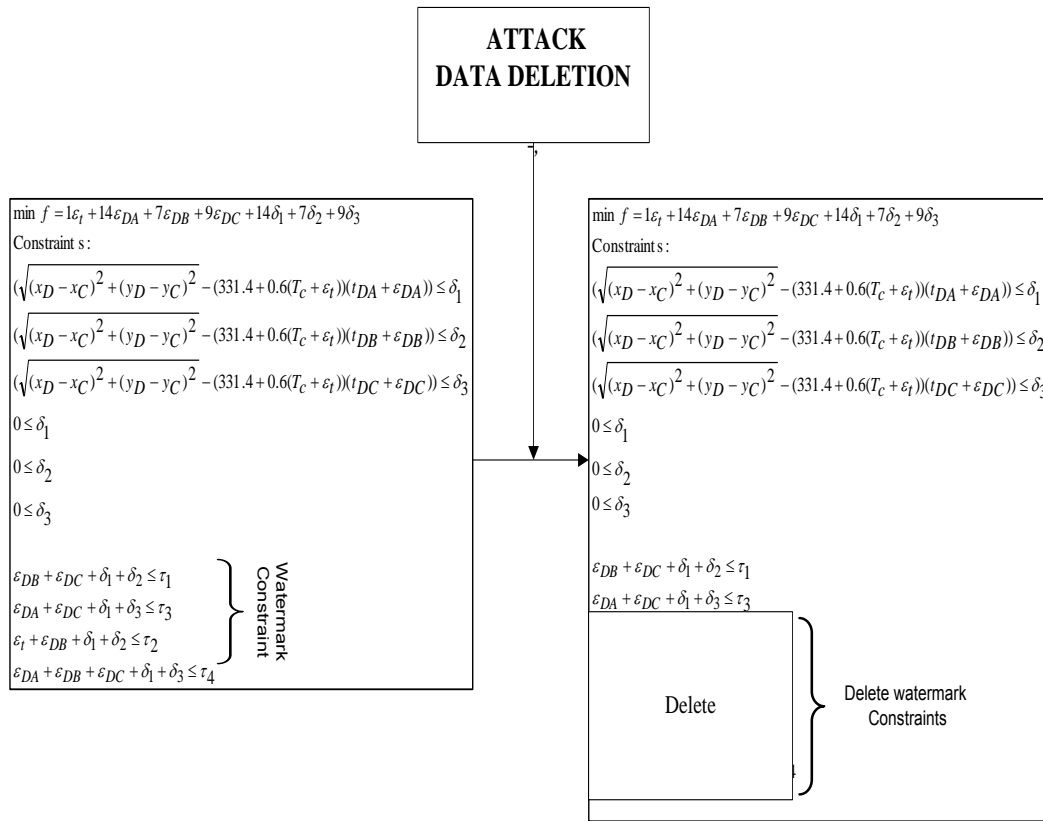


Figure 5.22 Deletion of watermark constraint data from the cover medium

5.5.3.3 Data Replication Attack

In this attack, the attacker adds some new constraints to the cover medium by replicating the existing constraints. These new replicated constraints can seriously hamper the cover medium’s performance: the new results of errors of cover medium cannot be approximated to the results of errors of cover medium before the attack. Particular parts of the network can easily be manipulated, or even completely disconnected by the attacker, by inserting the replicated constraints along with the new constraints into the existing constraints. Figure 5.23 shows replication of watermark constraint data from the cover medium.

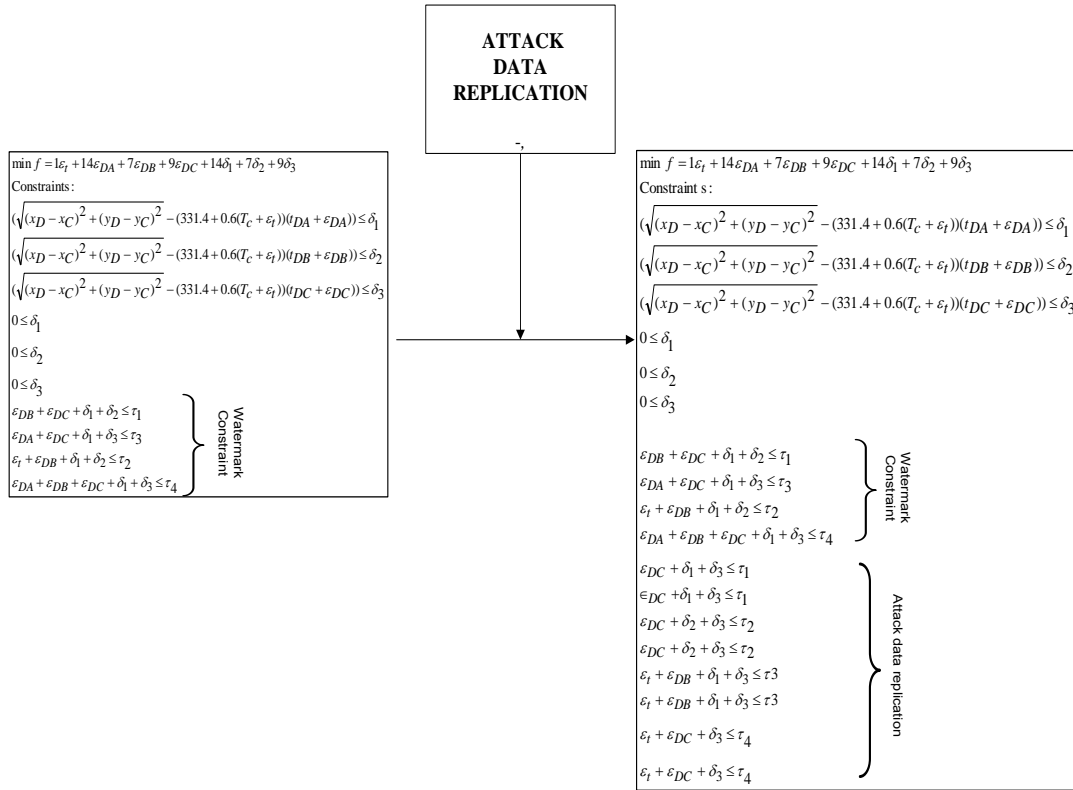


Figure 5.23 Watermark constraints data replication attack in the cover medium

5.5.3.4 Sybil Data Attack

The rapid growth of communication networks, such as the Internet and WSNs, has spurred the development of numerous collaborative applications. Reputation network system plays a pivotal role in these application by enabling multiple parties to establish relationships for mutual benefits. A Sybil data attack occurs when the attacker creates multiple identities and exploits them in order to manipulate a reputation score. The manipulated reputation score is computed and published by a reputation network system for a set of objects within a community using a WSN. A Sybil data attack occurs when an unauthorized attacker takes on multiple data identities. It subverts a reputation network system by introducing more than one identity constraints in the cover medium. The attacker uses these identities to acquire a larger influence than what he would normally have. A reputation network system becomes vulnerable to a Sybil attack depending on many factors. For example, the ease of generating constraint identities, the inclination or otherwise of the reputation network system to accept input from entities without a chain of trust that links them to a trusted entity, and the tendency or otherwise of the reputation network system to treat all the entities

identically. The objective of the Sybil data attack here is to find the results of errors of cover medium. Figure 5.24 shows the watermark constraints in the cover medium in Sybil data attack.

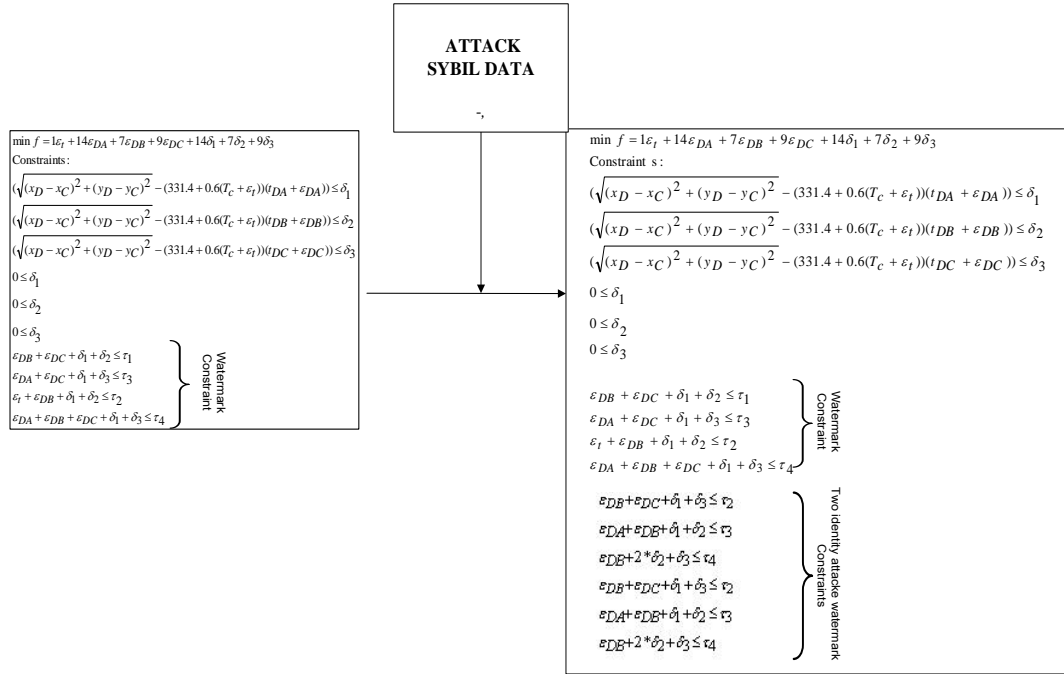


Figure 5.24 Watermark constraints in the cover medium in Sybil data attack

5.6 Results and Observations

This section discusses the experiments conducted and results observed on running the prototype. This is the sixth stage of the conceptual process described in Section 4.4.6.

First, we provide the results of the message sensed data embedding process. This process was undertaken by an embedder. Figure 5.25 shows the screenshot of the watermark embedding process.

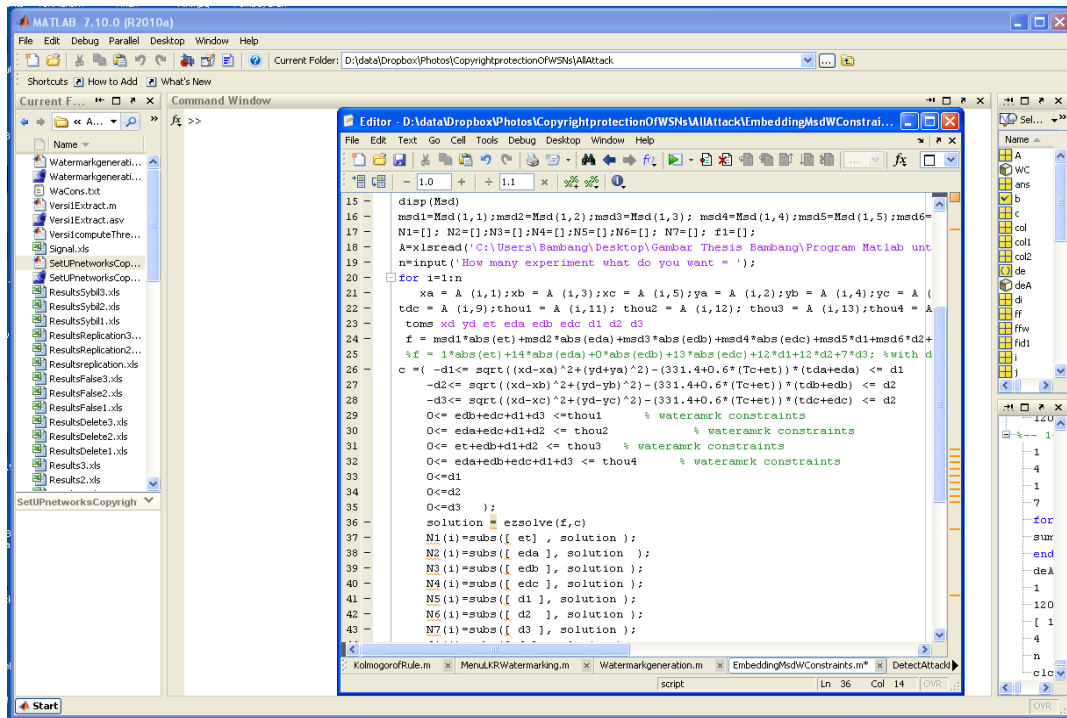


Figure 5.25 Screenshot of watermark embedding process Matlab Code

In this process, we use the first row of Table 5.3 as an input to the cover medium (NLSP) as follows:

Xa	Ya	Xb	Yb	Xc	Yc	Tda
475.064	351.37	242.991	347.2836	228.233	478.421	0.77162
6		2		8	7	5

Tdb	Tdc	Temp	Delta1	Delta2	Delta3	Theta1
0.10679	0.09282	57.6720	0.833677	0.18659	0.03997	0.94113
2		7	4	9	7	

Theta2	Theta3	Theta4
0.26914	0.15718	0.03181
9	9	7

The cover medium is as follows:

```
toms xd yd et eda edb edc d1 d2 d3
f =
msd1*abs(et)+msd2*abs(eda)+msd3*abs(edb)+msd4*abs(edc)+msd5*d1+msd6*d2+ms
d7*d3
c = {
-d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))* (tda+eda) <= d1
-d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))* (tdb+edb) <= d2
-d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))* (tdc+edc) <= d2
0<=d1
0<=d2
0<=d3 };
```

```
solution = ezsolve(f,c)
```

To solve this equation we use TOMLAB, and get following results:

```
=====
      MAIN MENU GENERATE WATERMARK
      LKR WATERMARKING TECHNIQUE
      1. Embedding watermark message sensed data ( MSD )
      2. MenuLKR Watermarking technique
=====
Message Sensed Data(MSD) is =
      1      14      7      9      14      7      9
What do you want the input data which you will compute by original
problem = 1
Problem type appears to be: lpcon
Time for symbolic processing: 0.44676 seconds
Starting numeric solver
===== * * *
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
Problem: --- 1: Problem 1          f_k          19.202759341488743000
              constr|)          0.000000819372765304
              f_k) + sum(|constr|) 19.202760160861509000
              f(x_0)             0.00000000000000000000

Solver: snopt.  EXIT=0.  INFORM=1.
SNOPT 7.2-5 NLP code
Optimality conditions satisfied

FuncEv    1 ConstrEv    22 ConJacEv    22 Iter    20 MinorIter    48
CPU time: 0.031200 sec. Elapsed time: 0.032000 sec.

solution =
      d1: 0
      d2: 0
      d3: 0
      eda: 1.1349
      edb: 5.5511e-017
      edc: 0.3683
      et: 0
      symb5: 0.3683
      symb6: -5.5511e-017
      symb7: 0
      symb8: 1.1349
      xd: 261.8514
      yd: 313.0482
```

et	eda	edb	edc	d1	d2	d3
0	1.13489	5.55112e-017	0.368257	0	0	0

We next compute all the input of Table 5.3. The results obtained are shown in Table 5.5 while Figure 5.26 shows the behaviour of error in the measurement of temperature, measurement of timer, and the measurement of error between the Euclidean measurement and that measured using TDoA.

Table 5.5 The results of error for message sensed data in WSN

No	et	eda	edb	edc	d1	d2	d3
1	0	1.134889	5.55E-17	0.368257	0	0	0
2	0	0.391018	1.001159	0.721391	0	0	0
3	0	0.694277	0.551107	2.22E-16	0	-2.8E-14	0
4	0	0	1.360715	0	0	0	0
5	0	0.571871	0.30367	0.125314	0	1.42E-14	0
6	0	0.202106	0.977522	3.45E-16	0	0	0
7	0	0.497659	1.35E-13	0.375235	-5.7E-14	0	0
8	0	0.96722	-6.7E-16	0.140027	0	0	0
9	0	0.720818	1.11E-16	0.610208	0	0	0
10	0	-6.9E-18	1.348207	0	0	9.73E-16	0
11	0	0.400313	0.776575	-2.1E-16	5.68E-14	0	0
12	0	0.374841	0.971116	0.929803	0	5.68E-14	0
13	0	0	0.188626	3.37E-14	0	8.48E-17	0
14	0	0	0	0.830486	0	0	0
15	0	0.449321	1.416444	1.31E-13	0	0	0
16	0	1.39E-13	-0.32752	-2.8E-17	0	0	0
17	0	2.78E-17	-0.61071	-8.3E-17	0	-7.1E-15	0
18	0	0.263067	0.852615	0	0	0	0
19	0	0.420564	0	1.134938	0	-1.4E-14	0
20	0	0	0	0.724389	0	-2.8E-14	0
21	0	0.066497	0	0.519879	0	0	0
22	0	0.095474	0.491735	0	0	0	0
23	0	0.758936	0.104547	1.29E-16	0	0	0
24	0	0	0.213606	0	0	0	0
25	2.78E-17	0.965798	0.92768	0	0	9.73E-16	0
26	0	0	-0.11404	0.838782	-2.8E-14	0	0
27	0	0.732052	1.46E-13	0.684669	0	0	0
28	0	0.030203	1.002383	1.138815	-2.8E-14	0	0
29	0	0.200279	0.02728	6.67E-14	0	0	0
30	0	-1.6E-14	0.420581	-1.4E-13	0	0	0
31	0	0.10411	0.397249	-5.1E-17	0	0	0
32	0	0.570101	0	0.349062	0	-2.8E-14	0

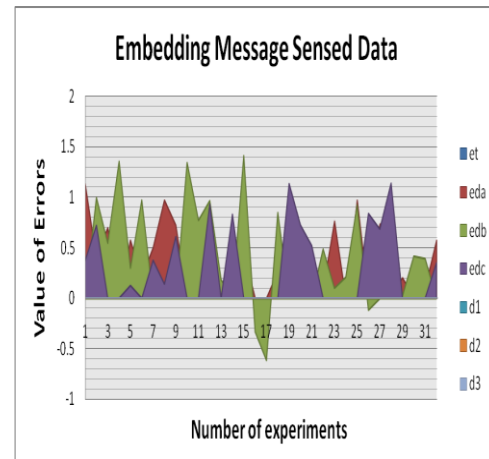


Figure 5.26 The value of error in the cover medium process

In the next step, we provide the results of the message sensed data and watermark constraints embedding process. The cover medium consists of the watermark constraints and the message sensed data.

```

toms xd yd et eda edb edc d1 d2 d3
f =
msd1*abs(et)+msd2*abs(eda)+msd3*abs(edb)+msd4*abs(edc)+msd5*d1+msd6*d2+msd7*d3;
c = { -d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))* (tda+eda) <= d1
-d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))* (tdb+edb) <= d2
-d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))* (tdc+edc) <= d2
0<= edb+edc+d1+d3 <= thou1
0<= eda+edc+d1+d2 <= thou2
0<= et+edb+d1+d2 <= thou3

```

```

0<= eda+edb+edc+d1+d3 <= thou4
0<=d1
0<=d2
0<=d3 };
solution = ezsolve(f,c)
    
```

To solve this equation (5.2), we use TOMLAB, and get the following results:

```

=====
                MAIN MENU GENERATE WATERMARK
                LKR WATERMARKING TECHNIQUE
                1. Embedding MSD and Watermark Constraints
                2. Menu LKR Watermarking technique
=====
Message Sensed Data(MSD) is =
      1      14      7      9      14      7      9

Problem type appears to be: lpcon
Time for symbolic processing: 0.57675 seconds
Starting numeric solver
===== * * *
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
==
Problem: --- 1: Problem 1
                f_k      12.790230275978020000
                sum(|constr|) 225.412070598005900000
                f(x_k) + sum(|constr|) 238.202300873983920000
                f(x_0)      0.00000000000000000000

Solver: snopt. EXIT=4. INFORM=13.
SNOPT 7.2-5 NLP code
Nonlinear infeasibilities minimized
Optimality conditions satisfied
FuncEv 1 ConstrEv 48 ConJacEv 48 Iter 31 MinorIter 100
CPU time: 0.062400 sec. Elapsed time: 0.047000 sec.
solution =
    d1: 0
    d2: 0
    d3: 0
    eda: 0.6203
    edb: -0.0927
    edc: 0.2657
    et: 1.0656
    symb10: 0.0927
    symb11: 1.0656
    symb12: 0.6203
    symb9: 0.2657
    xd: 248.0065
    yd: 348.4703
    
```

et	eda	edb	edc	d1	d2	d3
1.06556	0.620304	-0.0927369	0.265695	0	0	0

We next compute all the input of Table 5.3. The results obtained are shown in TTable 5.6 while Figure 5.23 shows the behaviour of error in the measurement of temperature measurement of timer, and the measurement of error between the Euclidean measurement and that measured using TDoA.

Table 5.6 The results of error for message sensed data and watermark constraints in WSNs

No	et	eda	edb	edc	d1	d2	d3
1	1.06556	0.620304	-0.09274	0.265696	0	0	0
2	-0.52494	0.122299	0.500279	-0.35519	0	0.953878	0
3	0.629591	0.681904	0.056409	-0.50963	0	0	0.453221
4	-0.58541	0.444767	0.671375	-0.44477	0	-2.1E-14	0
5	0.067165	0.550297	0.296991	-0.42418	0	0	0.127186
6	-0.24798	0.523544	0.370214	-0.37021	0	0.50959	0
7	0.851114	0.434698	0.006405	-0.06137	0	0	0.054963
8	0.270709	0.164214	-0.0733	0.0733	0	0.458848	0
9	0.26987	0.254104	-0.13644	0.27569	0	0	0
10	-5.6E-17	0.377486	0	-0.01221	2.01E-16	1.48E-14	0.012206
11	-0.20462	0.380408	0.419108	-0.33704	0	0	0
12	-0.63957	-4.2E-17	0.730334	0.094724	0	0	0
13	0	0	0.188626	-1.4E-13	3.34E-17	3.05E-15	0
14	0.963822	0.032656	-0.43549	0.377661	0	0	0.05783
15	-2.08974	0	0.987546	-0.80834	0	1.36285	0
16	0.184704	1.39E-13	-0.1847	0.184704	0	1.25E-16	0
17	0.330446	0	-0.33045	0.330446	0	0	0
18	0.59158	0.979257	0.173305	-0.8579	0	0	0.684596
19	0.173853	-0.55619	-0.28419	0.857827	0	0.521887	0
20	0.940071	0.714153	-0.43632	-0.43025	0	0	0.866564
21	0.671366	0.541064	-0.5725	-0.13636	0	0	0.708862
22	0	0.316082	0.259801	-0.15585	0	5.88E-15	0
23	0.086732	0.265111	0.044291	-0.0681	0	0	0.023812
24	-0.21223	0	0.158153	-0.05521	3.22E-15	0.055206	0
25	-0.97691	0.577727	0.600347	-0.53545	0	0.586062	0
26	1.25939	-0.40068	-0.559	0.486362	0	0	0.473322
27	0.716115	0.21234	-0.03621	0.282922	0	0	0
28	-0.45991	0.1611	0.729056	0.003111	0	0	0
29	0	0.200281	0.027276	0	0	1.45E-14	0
30	0	0.224778	0.196391	-0.18902	0	-8.9E-15	0
31	-0.53937	0.14322	0.199194	-0.19919	0	0.435961	0
32	0.383394	0.139942	-0.17635	0.176348	0	0.473765	0

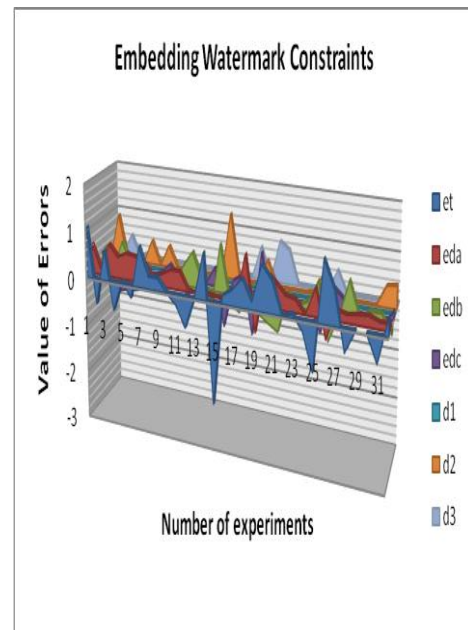


Figure 5.27 The value of error in watermark constraints embedding process

We next consider and evaluate the corresponding attack for LKR watermarking technique that can be used by the attacker in detail. There are four kinds of watermark constraint attacks:

5.6.1.1 False Data Insertion

As pointed out in Section 5.5.3.1, in this type of attack the attacker creates a number of different watermark constraints generated by the LFSR, hoping for new results of error that will map into the cover medium. The process of watermark constraint data insertion can be seen as follows:

```

=====
LKR WATERMARKING TECHNIQUE
COPYRIGHT OF SCALAR DATA IN WSNs
1. DETECT FALSE DATA INSERTION
2. Menu LKR Watermarking technique
=====
What Do you want = 1
Message Sensed Data (MSD) is =
  1   14   7   9   14   7   9

What do you want the input data which you will compute by original
problem = 1
Problem type appears to be: lpcon
Time for symbolic processing: 1.8735 seconds
Starting numeric solver

===== * * *
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
Problem: --- 1: Problem 1          f_k      227.612639313463430000
          m(|constr|)              0.000033961644652664
          x_k) + sum(|constr|)      227.612673275108080000
          f(x_0)                   0.00000000000000000000

Solver: snopt.  EXIT=0.  INFORM=1.
SNOPT 7.2-5 NLP code
Optimality conditions satisfied

FuncEv    1  ConstrEv    23  ConJacEv    23  Iter     20  MinorIter    62
CPU time: 0.265202 sec. Elapsed time: 0.796000 sec.
solution =

      d1: 6.5226e-015
      d2: 0
      d3: 0
      eda: 0.7127
      edb: -0.0927
      edc: 0.1733
      et: 215.4258
      symb1: 0.1733
      symb2: 0.0927
      symb3: 215.4258
      symb4: 0.7127
      xd: 249.8575
      yd: 348.4319

```

et	eda	edb	edc	d1	d2	d3
215.426	0.712744	-0.0927369	0.173255	6.52256e-015	0	0

We next compute all the input data of Table 5.3 to get the results of the error for false data attack which can be seen in Table 5.7, and the value of the error for false data attack which is depicted in Figure 5.28. The figure shows the behaviour of error in the measurement of temperature

measurement of timer, and the measurement of error between the Euclidean measurement and that measured using TDoA.

Table 5.7 The results of error for false data attack in WSNs

No	et	eda	edb	edc	d1	d2	d3
1	215.4257683	0.712744	-0.09274	0.173255	6.52E-15	0	0
2	683.9116461	0.407364	-0.024	-0.11598	0.139977	0	0
3	250.2199794	0.423637	0.337342	-0.25136	0	2.75E-15	0
4	65.63663329	0.08596	0.707953	-0.6803	0	0.594343	0
5	197.5651057	0.235164	0.252738	-0.10904	-7.67E-15	9.73E-16	0
6	87.56110409	0.391091	0.492773	-0.36032	1.84E-15	3.36E-15	0
7	209.582751	0.179556	-0.01857	0.193774	-3.99E-14	0	0
8	266.1542238	0.410235	-0.20672	-0.0393	0.246022	4.86E-16	4.86E-16
9	417.2520549	0.133427	-0.07758	0.216829	0	-2.76E-15	0
10	0	0.0534	0.0534	-0.0534	3.31E-14	-3.92E-15	0
11	164.741957	0.181636	0.396524	-0.13827	-8.28E-15	0	0
12	1223.453485	0	0.188919	0.094724	0	0	0
13	0	0	0.188626	-3.23E-27	-7.52E-15	-4.73E-16	0
14	148.2306176	0	-0.40085	0.410317	-2.41E-16	0	0
15	1575.326779	-0.04036	0.179209	-0.48023	4.45E-14	0.219567	0.301017
16	0	-4.23E-26	-0.18472	0.184718	1.12E-15	-8.62E-16	-2.58E-25
17	0	1.45E-16	-0.33045	0.330451	-3.75E-16	0	0
18	39.96658936	0.14779	0.818266	-0.02643	-2.64E-14	0	0
19	1247.344687	0.199257	-0.28263	0.100824	0.181803	0	0
20	261.7026365	-0.02587	-0.28391	0.309778	0.025872	0	0
21	84.78883309	0	-0.05288	0.404699	-9.69E-27	0	0
22	0	0.31608	0.259805	-0.15585	4.22E-15	0	0
23	402.1258202	0.131023	0.02465	0.065985	-7.56E-15	0	0
24	0	-2.63E-16	0.158157	-0.05521	1.53E-15	0.05521	0
25	745.7549265	0.209496	0.438836	-0.18935	0	1.07E-14	0
26	193.4725294	-0.42299	-0.08568	0.508672	0.422992	0	0
27	565.1598523	0.174164	-0.07438	0.321098	1.54E-14	0	0
28	1181.635631	-4.34E-19	0.174934	0.164212	-5.14E-16	0	0
29	0	0.200281	0.027276	0	-4.57E-15	0	0
30	0	0.224777	0.196392	-0.18902	0	-1.30E-14	0
31	124.6190228	0.093448	0.14322	-0.14556	-1.96E-15	0.049772	0.00234
32	214.3923935	0.410376	-0.28857	0.018135	0.270435	0	0

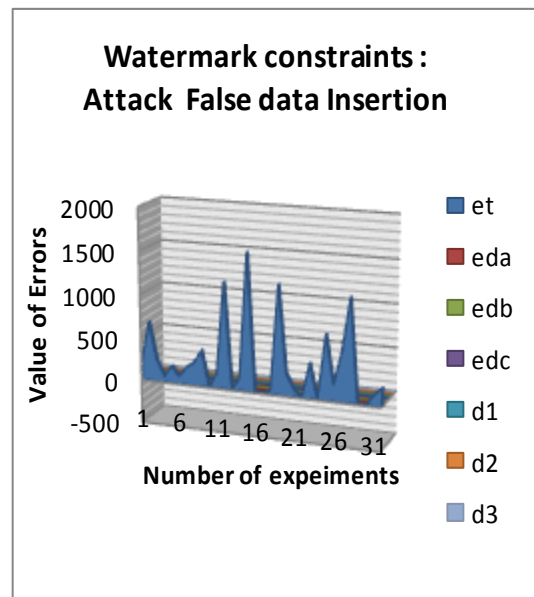


Figure 5.28 The value of error for false data attack in WSNs

The robustness can be computed as follows:

- ```

=====
LKR WATERMARKING TECHNIQUE
1. Detecting Process False Insertion
2. Menu LKR Watermarking technique
=====

```

CASES of FALSE DATA INSERTION

```

The value of threshold is 7.69725

The value of similarity is 10566

Whether the watermark constraints changes or not

Attack False data insertion change watermark constraints
because 7.697 < 1.057e+004
NO Robust against watermark constraints attacks false data Insertion

```

### 5.6.1.2 Data Deletion

As discussed in Section 5.5.3.2, in this type of attack the attacker deletes a number of watermark constraints generated by the LFSR, hoping that the new results of error will not approximate to the results of error of the cover medium without attack. The process of watermark constraint data deletion can be seen as follows:

```

=====
LKR WATERMARKING TECHNIQUE
COPYRIGHT OF SCALAR DATA IN WSNs
1. DETECT DATA DELETION
2. Menu LKR Watermarking technique
=====
What Do you want = 1
Read Message Sensed Data (MSD) is = 1 14 7 9 14 7
9
How many experiment what do you want = 32
Problem type appears to be: lpcon
Time for symbolic processing: 0.53121 seconds
Starting numeric solver
=====
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
=
Problem: --- 1: Problem 1 f_k 226.318410905678920000
 (|constr|) 0.000395269642641704
 f(x_k) + sum(|constr|) 226.318806175321550000
 f(x_0) 0.00000000000000000000

Solver: snopt. EXIT=0. INFORM=1.
SNOPT 7.2-5 NLP code
Optimality conditions satisfied

FuncEv 1 ConstrEv 30 ConJacEv 29 Iter 24 MinorIter 102
CPU time: 0.031200 sec. Elapsed time: 0.031000 sec.

solution =
d1: 0
d2: -2.0247e-015
d3: 4.8647e-016
eda: 0.8116
edb: 0
edc: 0.0744
et: 214.2862

```

---

```
symb1421: 0.0744
symb1422: 0
symb1423: 214.2862
symb1424: 0.8116
 xd: 252.2038
 yd: 399.2915
```

| et       | eda      | edb | edc      | d1 | d2     | d3       |
|----------|----------|-----|----------|----|--------|----------|
| 214.2862 | 0.811643 | 0   | 0.074356 | 0  | -2E-15 | 4.86E-16 |

We next compute all the inputs of Table 5.3 to get the results of the error for false data attack which can be seen in Table 5.8, and the value of the error for false data attack which is depicted in Figure 5.29. The figure shows the behaviour of error in the measurement of temperature measurement of timer, and the measurement of error between the Euclidean measurement and that measured using TDoA.

Table 5.8 The results of error for data deletion attack in WSNs

| No | et          | eda      | edb      | edc       | d1        | d2        | d3        |
|----|-------------|----------|----------|-----------|-----------|-----------|-----------|
| 1  | 214.286123  | 0.811657 | 0        | 0.074342  | 0.00E+00  | -2.81E-14 | 0         |
| 2  | 196.356407  | 1.008056 | 0.432157 | -0.28707  | 0         | 2.91E-15  | 0         |
| 3  | 250.2200278 | 0.423637 | 0.337342 | -0.25136  | 0         | 0.00E+00  | 0         |
| 4  | 0           | 0.443996 | 0.670604 | -0.444    | 0         | -1.50E-14 | 0         |
| 5  | 197.5651067 | 0.235164 | 0.252738 | -0.10904  | 0.00E+00  | 0.00E+00  | 7.77E-15  |
| 6  | 87.56106358 | 0.707139 | 0.176672 | -0.04422  | 0.00E+00  | 3.07E-14  | 0         |
| 7  | 209.5824776 | 0.179556 | -0.01857 | 0.193774  | 0.00E+00  | 0         | 0         |
| 8  | 121.915138  | 0.552036 | 0        | 0.144326  | -2.00E-14 | 1.94E-14  | -2.61E-13 |
| 9  | 288.5983986 | 0.358898 | -0.03165 | 0.170896  | 0         | 0.00E+00  | 0         |
| 10 | 0           | 0.386347 | 9.73E-16 | 0         | 0.00E+00  | -3.48E-16 | -9.73E-16 |
| 11 | 164.741958  | 0.181636 | 0.396524 | -0.13827  | 0.00E+00  | 0         | 0         |
| 12 | 1223.453424 | 0        | 0.188918 | 0.094724  | 0         | 0         | 0         |
| 13 | 0           | 0        | 0.188626 | -6.46E-27 | 0.00E+00  | 0.00E+00  | 0         |
| 14 | 148.2306216 | 0        | -0.40085 | 0.410317  | 0.00E+00  | 9.54E-18  | 2.85E-14  |
| 15 | 17.56168173 | 1.27138  | 0.957459 | -0.71687  | 0.00E+00  | 3.51E-14  | 0         |
| 16 | 0           | 1.39E-13 | -0.18472 | 0.184718  | 0.00E+00  | 0.00E+00  | 0.00E+00  |
| 17 | 0           | 0.00E+00 | -0.33045 | 0.330451  | 7.18E-16  | 3.58E-16  | 4.86E-16  |
| 18 | 39.96603912 | 0.147807 | 0.818283 | -0.02645  | 0.00E+00  | 0         | -4.95E-15 |
| 19 | 375.1667251 | 0.685773 | 3.47E-18 | 0.137753  | 0         | 0         | 0         |
| 20 | 231.7897733 | 0        | -0.36011 | 0.283906  | 0         | 0         | 0.076204  |
| 21 | 84.78883197 | 0        | -0.05288 | 0.404699  | 0.00E+00  | 0         | 0         |
| 22 | 0           | 0.149894 | 0.414134 | 1.44E-13  | -1.25E-15 | -1.37E-14 | 0         |
| 23 | 402.1257774 | 0.156215 | 0        | 0.040793  | 0.00E+00  | 0         | 0         |
| 24 | 0           | 0.00E+00 | 0.158157 | -0.05521  | -2.98E-15 | 0.05521   | 0         |
| 25 | 135.9292721 | 1.004362 | 0.625509 | -0.37602  | -4.86E-15 | 2.44E-15  | 0         |
| 26 | 308.978118  | 0        | -0.63076 | 0.08568   | 0         | 0         | 0.545083  |
| 27 | 565.1598536 | 0.174164 | -0.07438 | 0.321098  | 0.00E+00  | 0         | 0         |
| 28 | 1181.635614 | 0.00E+00 | 0.174935 | 0.164212  | 2.53E-15  | 0         | 0         |
| 29 | 0           | 0.200276 | 0.027285 | 0         | 2.28E-15  | 0         | 0         |
| 30 | 0           | 0.224778 | 0.196391 | -0.18902  | 0         | 2.29E-15  | 0         |
| 31 | 0           | 0.104109 | 0.39725  | 1.40E-13  | 1.18E-14  | 1.93E-14  | 0         |
| 32 | 44.01636979 | 0.502506 | -0.04136 | 0.287548  | 0         | -8.05E-16 | 0         |

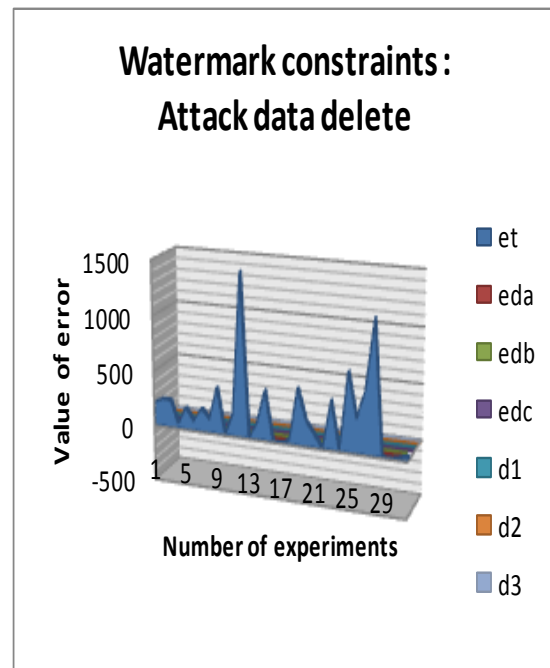


Figure 5.29 The value of error data deletion attack in WSNs

The robustness can be computed as follows:

```

=====
LKR WATERMARKING TECHNIQUE
1. Detecting Process attack data deletion
2. Menu LKR Watermarking technique
=====

The value of threshold is 7.69725

```



```

The value of similarity is 8208.81

Whether the watermark constraints changes or not

Attack False data deletion change watermark constraints
because 7.697 < 8209
NO Robust against watermark constraints attacks data deletion

```

### 5.6.1.3 Data replication

As discussed in Section 5.5.3.3, in this type of attack, the attacker seeks to add new constraints to the cover medium by replicating the existing constraints generated by the LFSR, hoping that the new results of error of the cover medium will not approximate to the results of error of the cover medium before attack. The process of watermark constraint replication can be seen as follows;

```

=====
LKR WATERMARKING TECHNIQUE
COPYRIGHT OF SCALAR DATA IN WSNs
1. DETECT REPLICATION ATTACK
2. Menu LKR Watermarking technique
=====
What Do you want = 1
Message Sensed Data (MSD) is =
 1 14 7 9 14 7 9

Problem type appears to be: lpcon
Time for symbolic processing: 0.75766 seconds
Starting numeric solver
===== * * *
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
==
Problem: --- 1: Problem 1 f_k 12.919039775614825000
 sum(|constr|) 241.222696246146140000
 f(x_k) + sum(|constr|) 254.141736021760980000
 f(x_0) 0.00000000000000000000

Solver: snopt. EXIT=4. INFORM=13.
SNOPT 7.2-5 NLP code
Nonlinear infeasibilities minimized
FuncEv 1 ConstrEv 19 ConJacEv 19 Iter 15 MinorIter 45
CPU time: 0.046800 sec. Elapsed time: 0.031000 sec.
solution =

 d1: 0
 d2: 0.0464
 d3: 0
 eda: 0.7933
 edb: -0.0464
 edc: 0.0464
 et: 0.7469
 symb1805: 0.0464
 symb1806: 0.0464
 symb1807: 0.7469

```

---

```
symb1808: 0.7933
xd: 264.2858
yd: 353.5212
```

| et          | eda         | edb         | edc      | d1 | d2          | d3       |
|-------------|-------------|-------------|----------|----|-------------|----------|
| 0.746893833 | 0.793262273 | -0.04636844 | 0.074356 | 0  | -0.04636844 | <b>0</b> |

We next compute all the input data of table 5.3 to get the results of error for false data attack which can be seen in Table 5.9, and the value of error for false data attack which is given in Figure 5.30. The figure shows the behaviour of error in the measurement of temperature measurement of the timer, and the error in measurement between the Euclidean measurement and that measured using TDoA.

Table 5.9 The results of the error data replication attack in WSNs

| No | et       | eda       | edb       | edc       | d1        | d2        | d3       |
|----|----------|-----------|-----------|-----------|-----------|-----------|----------|
| 1  | 0        | 1.134889  | 5.55E-17  | 0.368257  | 0.00E+00  | 0         | 0        |
| 2  | 0        | 0.391018  | 1.001159  | 0.721391  | 0         | 0         | 0        |
| 3  | 0        | 0.694277  | 0.551107  | 2.22E-16  | 0         | -2.84E-14 | 0        |
| 4  | 0        | 0         | 1.360715  | 0         | 0         | 0         | 0        |
| 5  | 0        | 0.571871  | 0.30367   | 0.125314  | 0.00E+00  | 1.42E-14  | 0        |
| 6  | 0        | 0.202106  | 0.977522  | 3.45E-16  | 0.00E+00  | 0.00E+00  | 0        |
| 7  | 0        | 0.497659  | 1.35E-13  | 0.375235  | -5.68E-14 | 0         | 0        |
| 8  | 0        | 0.96722   | -6.66E-16 | 0.140027  | 0         | 0.00E+00  | 0.00E+00 |
| 9  | 0        | 0.720818  | 1.11E-16  | 0.610208  | 0         | 0.00E+00  | 0        |
| 10 | 0        | -6.94E-18 | 1.348207  | 0         | 0.00E+00  | 9.73E-16  | 0        |
| 11 | 0        | 0.400313  | 0.776575  | -2.08E-16 | 5.68E-14  | 0         | 0        |
| 12 | 0        | 0.374841  | 0.971116  | 0.929803  | 0         | 5.68E-14  | 0        |
| 13 | 0        | 0         | 0.188626  | 3.37E-14  | 0.00E+00  | 8.48E-17  | 0        |
| 14 | 0        | 0         | 0         | 0.830486  | 0.00E+00  | 0         | 0        |
| 15 | 0        | 0.449321  | 1.416444  | 1.31E-13  | 0.00E+00  | 0         | 0        |
| 16 | 0        | 1.39E-13  | -0.32752  | -2.78E-17 | 0.00E+00  | 0.00E+00  | 0.00E+00 |
| 17 | 0        | 2.78E-17  | -0.61071  | -8.33E-17 | 0.00E+00  | -7.11E-15 | 0        |
| 18 | 0        | 0.263067  | 0.852615  | 0         | 0.00E+00  | 0         | 0        |
| 19 | 0        | 0.420564  | 0         | 1.134938  | 0         | -1.42E-14 | 0        |
| 20 | 0        | 0         | 0         | 0.724389  | 0         | -2.84E-14 | 0        |
| 21 | 0        | 0.066497  | 0         | 0.519879  | 0.00E+00  | 0         | 0        |
| 22 | 0        | 0.095474  | 0.491735  | 0         | 0.00E+00  | 0         | 0        |
| 23 | 0        | 0.758936  | 0.104547  | 1.29E-16  | 0.00E+00  | 0         | 0        |
| 24 | 0        | 0.00E+00  | 0.213606  | 0         | 0.00E+00  | 0         | 0        |
| 25 | 2.78E-17 | 0.965798  | 0.92768   | 0         | 0         | 9.73E-16  | 0        |
| 26 | 0        | 0         | -0.11404  | 0.838782  | -2.84E-14 | 0         | 0        |
| 27 | 0        | 0.732052  | 1.46E-13  | 0.684669  | 0.00E+00  | 0         | 0        |
| 28 | 0        | 3.02E-02  | 1.002383  | 1.138815  | -2.84E-14 | 0         | 0        |
| 29 | 0        | 0.200279  | 0.02728   | 6.67E-14  | 0.00E+00  | 0         | 0        |
| 30 | 0        | -1.57E-14 | 0.420581  | -1.39E-13 | 0         | 0.00E+00  | 0        |
| 31 | 0        | 0.10411   | 0.397249  | -5.07E-17 | 0.00E+00  | 0         | 0        |
| 32 | 0        | 0.570101  | 0         | 0.349062  | 0         | -2.84E-14 | 0        |

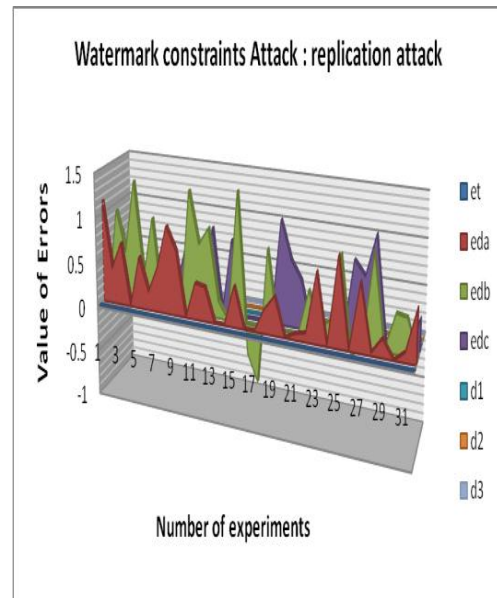


Figure 5.30 The value of error data replication attack in WSNs

The robustness can be computed as follows:

```

=====
LKR WATERMARKING TECHNIQUE
1. Detecting Process attack data replication
2. Menu LKR Watermarking technique

```

```

The value of threshold is 7.69725

The value of similarity is 1.03848

Whether the watermark constraints changes or not

```

Attack False data replication do not change the watermark  
 Because  $7.697 \Rightarrow 1.038$   
 Robust against watermark constraints attacks replication

### 5.6.1.4 Data Sybil

As discussed in Section 5.5.3.4, the attacker creates multiple identities and exploits them, in order to manipulate the reputation score computed by the reputation network system. Data Sybil is defined as a malicious device illegitimately taking on multiple data identities. The watermark constraints in data Sybil can be seen as follows:

```

=====
 LKR WATERMARKING TECHNIQUE
 COPYRIGHT OF SCALAR DATA IN WSNS
 1. DETECT DATA SYBIL
 2. Menu LKR Watermarking technique
=====
What Do you want = 1
Message Sensed Data (MSD) is =
 1 14 7 9 14 7 9

Problem type appears to be: lpcon
Time for symbolic processing: 0.95299 seconds
Starting numeric solver
=====
=====
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-
01
=====
==
Problem: --- 1: Problem 1 f_k 12.268989178202528000
 sum(|constr|) 481.983101398175900000
 f(x_k) + sum(|constr|) 494.252090576378410000
 f(x_0) 0.000000000000000000

Solver: snopt. EXIT=4. INFORM=13.
SNOPT 7.2-5 NLP code
Nonlinear infeasibilities minimized

FuncEv 1 ConstrEv 36 ConJacEv 36 Iter 25 MinorIter 68
CPU time: 0.078001 sec. Elapsed time: 0.047000 sec.

solution =
 d1: 0
 d2: 0.0464
 d3: 0
 eda: 0.6181
 edb: -0.0464
 edc: 0.2216
 et: 0.9728
 symb2189: 0.2216
 symb2190: 0.0464
 symb2191: 0.9728
 symb2192: 0.6181
 xd: 264.2864

```

---

yd: 353.5479

| et          | eda         | edb             | edc         | d1 | d2         | d3 |
|-------------|-------------|-----------------|-------------|----|------------|----|
| 0.972823517 | 0.618066216 | -<br>0.04636844 | 0.221564496 | 0  | 0.04636844 | 0  |

We next compute all the input data of Table 5.3 to get the results of error for data Sybil which can be seen in Table 5.10, and the value of error for data Sybil which is given in Figure 5.31. The figure shows the behaviour of error in the measurement of temperature measurement of timer, and error in measurement between the Euclidean measurement and that measured using TDoA.

---

Table 5.10 The results of error of Sybil data attack in WSNs

| No | et          | eda       | edb       | edc      | d1        | d2        | d3        |
|----|-------------|-----------|-----------|----------|-----------|-----------|-----------|
| 1  | 0.972823517 | 0.618066  | -0.04637  | 0.221564 | 0.00E+00  | 0.046368  | 0         |
| 2  | 0.869144262 | 0.267388  | -0.14725  | 0.147248 | 0         | 0.207318  | 0         |
| 3  | 0.653963033 | 0.681904  | 2.63E-17  | -0.54167 | 0         | 3.20E-02  | 0.541668  |
| 4  | 0.08596042  | 0.707953  | 2.63E-17  | -0.34839 | 0         | 0         | 0.348387  |
| 5  | 0.220462373 | 0.406603  | 0.143694  | -0.28048 | 0.00E+00  | 0.00E+00  | 0.280483  |
| 6  | 0.631822168 | 0.523544  | 0         | 0        | 0.00E+00  | 0.00E+00  | 0         |
| 7  | 0.85111486  | 0.435293  | 0.006405  | -0.06196 | 0.00E+00  | 0         | 0.061964  |
| 8  | 0.610813789 | 0.164214  | -0.07333  | 0.073327 | 0         | 1.19E-01  | 0.00E+00  |
| 9  | 0.133427403 | 0.254104  | -0.06822  | 0.207469 | 0         | 6.82E-02  | 0         |
| 10 | 0           | 0.377486  | -1.50E-18 | -0.01221 | -9.73E-16 | -5.14E-16 | 0.012206  |
| 11 | 1.29E-26    | 0.247987  | 0.21449   | -0.20462 | 0.00E+00  | 0         | 0.204618  |
| 12 | -1.50E-18   | 0         | 0.090759  | 0.094724 | 0         | 0         | 0         |
| 13 | 0           | -3.23E-27 | 0.188626  | 0.00E+00 | 1.42E-14  | 6.15E-15  | -1.42E-14 |
| 14 | 0.528831777 | 0.007032  | -0.12059  | 0.282694 | 0.00E+00  | 0.12059   | 0         |
| 15 | 0.171053852 | 0.179209  | 0         | 0        | 0.00E+00  | 0.089604  | 0         |
| 16 | 0           | 1.39E-13  | -0.06107  | 0.337073 | 0.00E+00  | 6.11E-02  | 0.00E+00  |
| 17 | 0           | -1.13E-01 | -0.27579  | 0.226331 | 0.00E+00  | 0.275789  | 0.162623  |
| 18 | 2.63E-17    | 0.214372  | 0.764885  | -0.09302 | 0.00E+00  | 0         | 0.093016  |
| 19 | 0.411555161 | 1.10E-13  | -0.28308  | 0.167719 | 0.132811  | 0.150265  | 0         |
| 20 | 0.503753573 | 0.076334  | -0.07791  | 0.129657 | 0         | 0.077914  | 0         |
| 21 | 0.098869176 | 0.069154  | 0         | 0.335545 | 0.00E+00  | 0         | 0         |
| 22 | 0.469516361 | 0.420039  | -6.50E-17 | -0.11652 | 0.00E+00  | 0         | 0.116518  |
| 23 | 0.086732282 | 0.265111  | 0.044291  | -0.0681  | 0.00E+00  | 0         | 0.068103  |
| 24 | 2.10E-20    | 2.17E-01  | 0.001126  | -0.21705 | 2.90E-15  | -8.02E-15 | 0.217049  |
| 25 | 0.195210577 | 0.628337  | 0.014285  | 0        | 0         | 1.45E-14  | 0         |
| 26 | 0.700387616 | -0.04284  | -0.04284  | 0.08568  | 0         | 0.04284   | 0         |
| 27 | 0.679907339 | 0.248548  | 0         | 0.246714 | 0.00E+00  | 0         | 0         |
| 28 | 1.57E-13    | 6.24E-01  | 0.26915   | -0.45991 | 0.00E+00  | 0         | 0.459906  |
| 29 | 0           | 0.20028   | 0.027277  | 0        | 0.00E+00  | 2.76E-14  | 0         |
| 30 | 0           | 0.413236  | 0.007369  | -0.27773 | 0         | 0.00E+00  | 0.277726  |
| 31 | -1.50E-18   | 0.14322   | -1.50E-18 | -0.03344 | 2.73E-02  | 0.068523  | 0.006173  |
| 32 | 0.680810812 | 0.139942  | -0.18459  | 0.162265 | 0.022324  | 0.162265  | 0         |

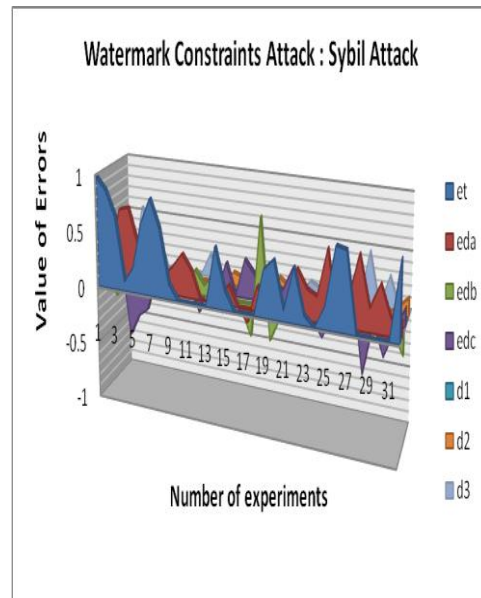


Figure 5.31 The value of error of Sybil data attack in WSNs

```

=====
LKR WATERMARKING TECHNIQUE
1. DETECT ATTACK SYBIL DATA
2. Menu LKR Watermarking technique
=====
The value of threshold is 7.69725

The value of similarity is 4.30362

Whether the watermark constraints changes or not

Attack data Sybil do not change the watermark
Because 7.697 => 4.304

```

---

Robust against watermark constraints attacks Sybil

## 5.7 Validation and discussion

Herein, we advanced a robust watermarking technique, termed LKR watermarking technique. The watermark constraints were embedded into the cover medium NLSP. The embedding process was based on equation constraints. The scalar decimal data was converted into binary bits. The LFSR was used to expand the binary bits into a sequence of binary sequence, using its characteristic polynomial. The bit numbers were matched with the corresponding variable numbers using the Kolmogorov rule, to generate the watermark constraints. If a variable within the group got the bit one assigned to it, the linear included it; on the other hand, if a variable within the group got the bit zero assigned to it, the linear did not include it. By testing several watermark constraint attacks, we found that the proposed technique works well in the cover medium NLSP. The robustness is achieved by embedding watermark constraints into NLSP. The normalized difference in error from the optimal solution between the watermarked solution and the solution obtained without watermark verifies that the watermark constraints are present. The threshold measure is given by the normalized correlation coefficient between the normalized difference error from the optimal solution between the watermarked solution  $X'$ , and the solution obtained without watermark.

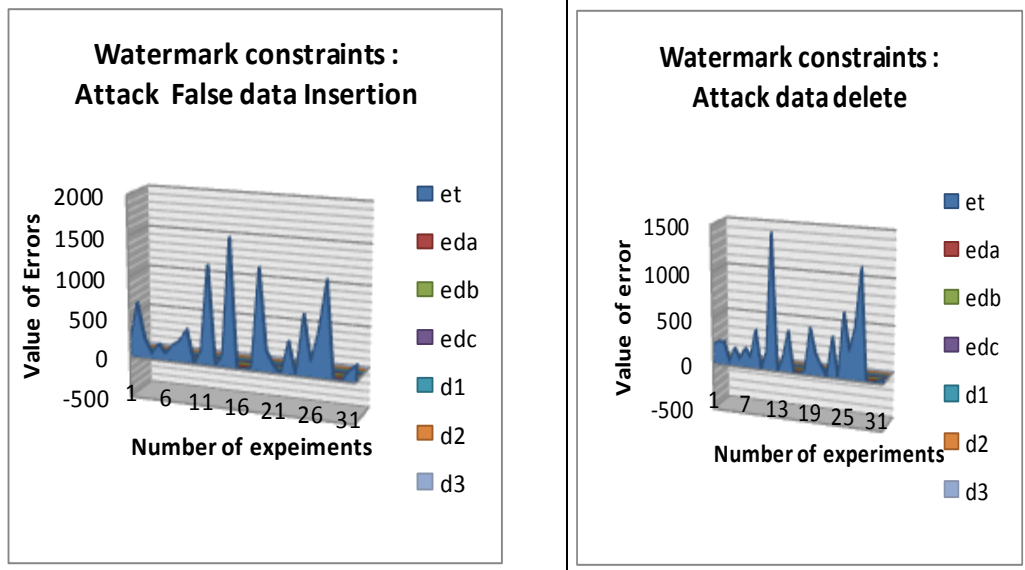
Threshold is measured by  $\frac{C * X'}{\sqrt{X' * X'}}$  where  $C = X' - X$ . The normalized coefficient of

correlation of the normalized difference error from the optimal solution between the watermarked solutions  $X'$ , and the solution obtained with watermark constraints  $X''$ , gives the similarity

measure. The similarity is measured by  $\frac{C' * X'}{\sqrt{X'' * X''}}$  where  $C' = X'' - X$ . If the value of threshold

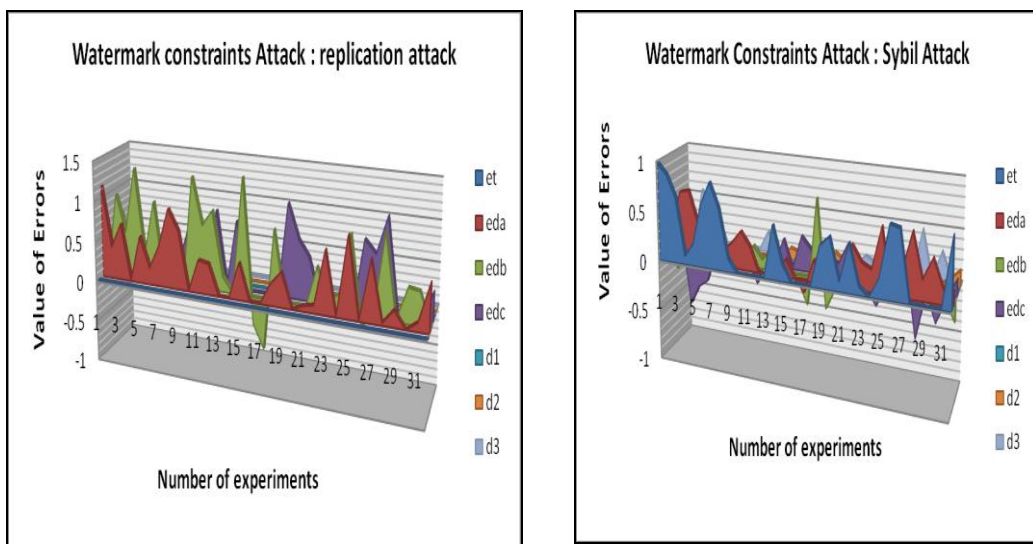
is greater than the value of similarity, it means the watermark constraints are robust against watermark constraint attacks.

---



a. False data insertion attack

b. Data deletion attack



c. Data replication attack

d. Sybil attack

Figure 5.32 All watermark constraint attacks

From the experiments conducted, following observations were made:

1. In false data insertion attack on watermark constraints, the false data insertion changes the watermark constraint (as shown in Figure 5.35. a) because  $7.697 < 1.057e+004$ . It means that the model is not robust against false data insertion attack.



2. In data deletion attack on watermark constraints, the data deletion changes the watermark constraints (as shown in Figure 5.35 b) because  $7.697 < 8209$ . It means that the model is not robust against data deletion attack.
3. In data replication attack on watermark constraints, the data replication does not change the watermark constraints (as shown in Figure 5.35 c) because  $7.697 > 1.038$ . It means that the model is robust against data replication attack.
4. In data Sybil attack on watermark constraints, the data Sybil does not change the watermark constraints (as shown in Figure 5.35 b) because  $7.697 > 4.304$ . It means that the model is robust against data Sybil attack.

Thus, the proposed scheme is shown to be robust against the replication of different watermark constraints and creation of multiple identities for watermark constraints. However, it is not robust against other watermark constraint attacks, such as insertion of one or more false watermark constraints and deletion of one or more watermark constraints.

The robustness of the proposed technique can be further improved by adding more constraints in the case of false data insertion attack, so that we get the value of similarity as 4.81673. This similarity is less than the threshold of 7.697. However, there are some weaknesses in the case of data deletion attack because, when we delete one and three watermark constraints, we get the value of similarity as 8208.81 and 317.16 respectively. These similarities are greater than the threshold of 7.697. One drawback of the proposed approach is that it needs not only scalar data but also images. This issue has been dealt with in Chapter 6, where we propose the GPKR watermarking technique.

## 5.8 Comparative analysis

This is the last stage of the conceptual process described in Section 4.5.6, where the proposed solution is compared with the existing solutions from the literature, as discussed in Chapter 2. Therefore, this chapter carries out a comparative analysis of this model with other such models proposed by different researchers. The results of this comparative analysis are given in Table 5.7.

Table 5.11 A comparative analysis with other approaches copyright protection in WSNs

| Kind of attacks      | Feng,J.P et al<br><br>(Jessica and Potkonjak 2003) | F Koushanfar, F et al<br><br>(Koushanfar and Potkonjak 2007) | Julia al bath et.al<br><br>(Xiangqian 2009) | Zhang et al.<br><br>(Zhang, Liu, and Das 2008), | Xiao et.al<br><br>(Rong, Xingming, and Ying 2008) | Xuejun et al<br><br>(Xuejun 2010) | Kamel et al.<br><br>(Kamel 2011) | Harjito, B |
|----------------------|----------------------------------------------------|--------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|---------------------------------------------------|-----------------------------------|----------------------------------|------------|
| False data insertion | x                                                  | x                                                            | x                                           | x                                               | x                                                 | x                                 | x                                | X          |
| Data deletion        | x                                                  | x                                                            | x                                           | x                                               | x                                                 | x                                 | √                                | x          |
| Packet replication.  | x                                                  | x                                                            | x                                           | x                                               | x                                                 | x                                 | x                                | √          |
| Sybil attack         | x                                                  | x                                                            | x                                           | x                                               | x                                                 | x                                 | x                                | √          |

√ provides copyright data protection , x does not provide copyright data protection

In this analysis, 8 approaches have been compared, in terms of false data insertion, data deletion, packet replication, and Sybil attack. Feng, J. P. et al. (Jessica and Potkonjak 2003), Koushanfar, F. et al. (Koushanfar and Potkonjak 2007), Julia Albath et Al. (Albath 2007), Zhang et al., Xiao et al. (Rong, Xingming, and Ying 2008), and Xuejun et al. (Xuejun 2010). Juma, Kamel and Kaya (2008) do not provide solution for data deletion, packet replication and Sybil attacks. Kamel et al. (Kamel and Juma 2011) provide solution for data deletion but not for packet replication and Sybil attacks. Our approach provides copyright data protection against packet replication and Sybil attacks.

In table 5.11 8 approaches have been compared: Albath, J. et al., Feng, J. P. et al., Koushanfar, F. et al., Zhang et al., and Xio et al. do not explain security attacks. The feasibility of false data insertion and data deletion can only be determined by the Forward Chaining Watermark (FWC) method proposed by Kamel et al. However this method only withstands data deletion, not false data insertion.

## 5.9 Conclusion

In this paper, we have proposed the LKR watermarking technique for copyright protection of data in WSNs. Our strategy aims at copyright protection of scalar data during transmission between sensor nodes in WSNs, against a variety of attacks, such as data deletion, data replication, data modification, Data Sybil attack, false data insertion, and selective forwarding. We have not discussed some types of attacks, such as physical attack, node malfunction and Denial of service attack. We have verified that our technique can protect copyright data against packet replication and Sybil attacks. However, it does not protect this data against false data insertion, data deletion, data modification and selective forwarding. Therefore, we still need to improve our technique considering various circumstances in which attackers launch different kinds of attacks for the future work.

---

---

# CHAPTER SIX

## GPKR WATERMARKING TECHNIQUE

This chapter presents:

- ▶ an introduction to the GPKR watermarking technique for copyright protection of images in WMSNs,
- ▶ a general overview of our GPKR watermarking technique for copyright protection of images in WMSNs,
- ▶ experimentation and testing of our GPKR watermarking technique for copyright protection of images in WMSNs,
- ▶ evaluation, validation and comparative study of our technique for copyright protection of images in WMSNs.

### 6.1 Introduction

This chapter addresses the problem of copyright protection of images by ensuring that the watermark is secure during transmission between sensor nodes in WMSNs. It considers the issue of copyright protection of images in WMSNs for various purposes, such as intellectual property protection. It presents a novel Gaussian Pyramids Kolmogorov Rule (GPKR) watermarking scheme that embeds watermark constraints in the cover medium NLSP. The novelty of this watermarking scheme lies in the use of watermark constraints derived from the sensed multimedia data by Gaussian Pyramids and Kolmogorov rule. We name this technique as GPKR watermarking technique.

### 6.2 The Proposed GPKR Watermarking Technique

This section provides a general overview of the GPKR watermarking technique as our solution for copyright protection of images, and then outlines the requirements to address the problem.

---

## 6.2.1 A General Overview of GPKR Watermarking Technique

WMSNs are an emerging type of sensor networks consisting of sensor nodes equipped with microphones, cameras, and other sensors that produce multimedia content. These networks have not only changed and improved the existing sensor applications, such as tracking and environment monitoring, they have also enabled several new applications ranging from military to modern healthcare. These networks collect and store multimedia data and then send them to other multimedia nodes or servers.

The new technologies allow validation of multimedia data during transit, but not after the data have reached their destination. One of the challenges with these technologies is to ensure that the source of the data, i.e. the image, is preserved even after it leaves the WMSN node. This is important as the image can be used by other unauthorized applications.

With the demanding constraints of nodes' limited computational capability, the key issue for WMSNs is designing viable security mechanisms to ensure confidentiality, integrity and authentication, to prevent malicious attacks. Besides the inherent limitations of communication and computing in WMSNs, the nature of their deployment makes them even more vulnerable to various attacks. Most of the prior works on securing sensor networks have used traditional security solutions based on public key cryptography algorithm and digital signatures. These techniques usually execute thousands of operations. Further, these security mechanisms are only useful in ensuring security during communication; once the data reach their destination and are decrypted, they can be used by anyone without permission. Thus, cryptography does not address the problem of copyright protection. Hence, research in the area of watermarking and WMSNs is becoming increasingly important. Watermarking technique is a technique traditionally used for providing copyright protection to multimedia data, like images and video clips. It can also be incorporated in WMSNs to ensure copyright protection of digital images and video footages.

## 6.2.2 Requirements

Following requirements are laid down for the proposed watermarking technique for copyright protection, in order to address the issue of copyright infringement of valuable sensory multimedia

---

---

data, like images. This represents the first stage of the conceptual process described in Chapter 1 in which the requirements are elicited and prioritized. The proposed algorithm is to be implemented on valuable sensory data; however, it can be extended to images and audio. The requirements are as follows:

1. Coverage: The GPKR algorithm should operate on a minimum of 50 nodes randomly deployed within 200 meter of length and 100 meter of width.
2. Cover medium: The GPKR algorithm should only use scalar data as cover medium generated through atomic trilateration process (shown in Figure 6.1).
3. Copyright protection: The GPKR algorithm should be able to ensure that the copyright is securely embedded in the cover medium.
4. Watermark constraints: The GPKR algorithm should use watermark constraints rather than a binary matrix watermark, because watermark constraints provide subjective detection.
5. Robustness: The watermark should be embedded robustly to handle watermark attacks.
6. Non-blind detection: The algorithm should offer non-blind extraction and detection.

Section 6.2.3 discusses the design rationale for copyright protection of images in WMSNs, incorporating each of the requirements outlined above. It also shows how each requirement has been addressed in the design decision. This concludes stage one of the conceptual process and paves the way for the next stage.

### **6.2.3 Design Rationale**

This sub-section proposes the design rationale for the GPKR watermarking technique to protect images in WMSNs, satisfying the requirements outlined in Section 6.2.2. This represents the second stage of the conceptual process described in Section 4.4.2 , in which all the requirements are addressed by finalising the design decision. We propose the following design decisions in the framework, as shown in Figure 6.1.

---

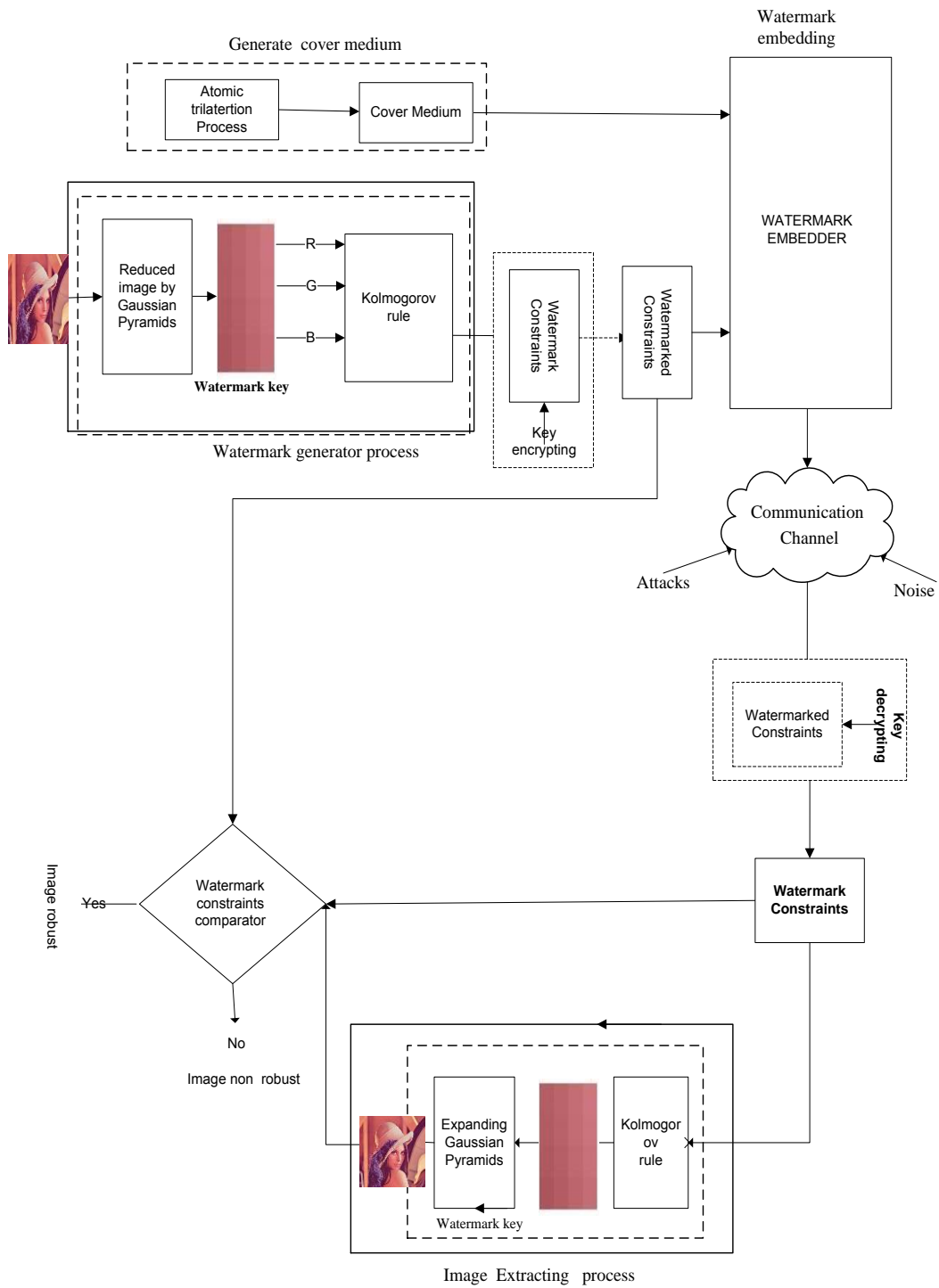


Figure 6.1 The general model of GPKR watermarking technique for copyright protection of images in WMSNs

1. The network setting consists of 50 sensor nodes randomly deployed within 200 meter of length and 100 meter of width. This network provides random positions for three multimedia sensor nodes used to start up the cover medium (Req. 1)
2. The cover medium is generated by using the theory of atomic trilateration as discussed in Section 0. This cover medium is used to insert watermark constraints (Req. 2, Req. 3).
3. The watermark constraints are generated using the Kolmogorov rule on the reduced image in an RGB colour (Req. 4).
4. Robustness of the watermark constraints is measured using the normalized difference error from the solution between the watermarked solution and the solution obtained without watermark (Req. 5).
5. The cover medium is required for the detection of watermark constraints The watermark constraints can be used to produce binary matrix, which is then converted into decimal matrix to get the reduced image. The sensory image is generated by expanding the reduced image (Req. 6).

### **6.3 Theoretical Foundation for GPKR Watermarking Technique.**

This sub-section proposes a theoretical foundation for our GPKR watermarking technique for copyright protection of images. This represents the third stage of the conceptual process described in Section 4.4.3 in which the design decisions are analysed and an algorithm presented.

The proposed scheme offers robustness against false data insertion and data replication. However it is not robust against data deletion and data modification, which are also not tackled in the literature (Honggang Wang, Dongming Peng, and Wei Wang 2008) (Pingping, Yao Jiangtao, and Zhang Ye 2009) (Wang 2010).

Looking at similar other schemes, Pingping, Yao Jiangtao, and Zhang Ye (2009) embed the watermark into the low-frequency coefficients of DCT. The embedding process consists of dividing the image into  $8 \times 8$  blocks, performing DCT of each block, selecting the coefficient at the

---



location  $(p, q)$  in each block of the blue component, and finally replacing the chosen coefficient with the new robust watermarked coefficient:  $DCT'(p, q) = \text{sign}(DCT(p, q)) + (\alpha * w)$ . This scheme is robust against cropping, jpg q=20, and paper salt noise.

Honggang Wang, Dongming Peng, and Wei Wang (2008), on the other hand, use adaptive watermarking positions dynamically chosen to embed watermark according to network conditions, to achieve both energy efficiency and security. The two adaptive thresholds  $(T1, T2)$  are used to filter and decide the appropriate embedding positions for watermarking. This scheme is robust against jpg compression, using the quality parameter 90. Meanwhile, Xiangjun, Shaodong, and Le (2008) embed the watermark into two adaptive thresholds  $(T1, T2)$  under the specified watermarking scheme. The two adaptive thresholds  $(T1, T2)$  are used to filter and decide the appropriate embedding positions for watermarks. The embedding positions are functions of these two thresholds. This scheme is robust against compression, using the quality parameter 90. The novel feature of the GPKR watermarking technique is that it uses Gaussian Pyramid Transforms for reducing the original image to get a reduced image, and then expand the reduced image to get the original image. It also uses Kolmogorov rule for numbering the variables of the linear equation to get watermark constraints. Thus, the GPKR watermarking technique uses Gaussian Pyramids and Kolmogorov rule for copyright protection of images, and hence the name GPKR.

The GPKR watermarking technique comprises four steps:

- Cover medium generation
- Watermark generation
- Watermark embedding
- Watermark extraction & detection

### 6.3.1 Cover Medium Generation

This sub-section explains the process of generating cover medium using the atomic trilateration process. The pseudo-code for generating cover medium is shown in Pseudo Code 6.1 and the flowchart in Figure 6.2.

---

## Pseudo Code 6.1 Generating cover medium

**Input:**

|                                               |                                                                                                                                             |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| $(x_A, y_A), (x_B, y_B), (x_C, y_C)$          | Position of two-dimensional three sensor network                                                                                            |
| $T_c$                                         | Temperature of the propagation media                                                                                                        |
| $t_{DA}, t_{DB}, t_{DC}$                      | Time for transmission between nodes D to A, D to B and D to C                                                                               |
| $V_s$                                         | Speed of the acoustic signal                                                                                                                |
| $\epsilon_t$                                  | Error in the measurement of temperature                                                                                                     |
| $\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$ | Error in the measurement of the timer from D to A, D to B and D to C                                                                        |
| $\delta_1, \delta_2, \delta_3$                | Error in the measurement between the Euclidean measurement and the measurement using time differences of optimal D to A, D to B, and D to C |
| $\delta_4$                                    | Auxiliary variables to the cover medium                                                                                                     |

**Output**

The cover medium

$$\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3 + \delta_4$$

Constraints

$$\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3$$

**Step 1: Getting all input data**

Position of the two-dimensional, three sensor network  $(x_A, y_A), (x_B, y_B)$ , and  $(x_C, y_C)$   
 Temperature of the propagation media ( $T_c$ ), Time for transmission between the nodes D to A, D to B, and D to C, Error in the measurement of temperature ( $t_{DA}, t_{DB}, t_{DC}$ ), Error in the measurement of the timer from D to A, D to B, and D to C ( $\epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$ ), and error in the measurement between the Euclidean measurement and the measurement using time differences of optimal D to A, D to B, and D to C ( $\delta_1, \delta_2, \delta_3$ ).

**Step 2: Computing the speed of acoustic signal and adding the error of temperature, as speed is one of the requirements for measuring the distance between two sensor nodes**

Compute the speed of acoustic signal using  $V_s = 331.4 + 0.6(T_c + \epsilon_t)$

**Step 3: Computing the distance between the node  $D$  and the sensor nodes A, B and C, using time differences of arrival (TDoA) and adding the error of measurement of the timer respectively**

Compute the distance of  $d_{DA} = V_s * (t_{DA} + \epsilon_{DA})$ ,  $d_{DB} = V_s * (t_{DB} + \epsilon_{DB})$ , and  $d_{DC} = V_s * (t_{DC} + \epsilon_{DC})$ .

**Step 4: Computing the distance between the node  $D$  and the sensor nodes A, B and C, using the Euclidean theorem**

Compute the distance of  $d_{DA} = \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}$ ,

$d_{DB} = \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2}$ , and  $d_{DC} = \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}$

**Step 5: Computing the difference between the distance using TDoA and the distance using the Euclidean theorem, and adding the error in the measurement between the Euclidean measurement and the measurement using time differences of optimal D to A, D to B, and D to C.**

Use step (2) into step (3), compute difference between step (4) and step (3) and add error between step (4) and step (3).

$$\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3$$

**Step 6: Computing the minimized error in the system of equations ( step 5)**

Compute the minimized objective function  $\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3 + \delta_4$

**Step 7: Generating cover medium**

Append step (5) and step (6) to get Non Linear System Programming.

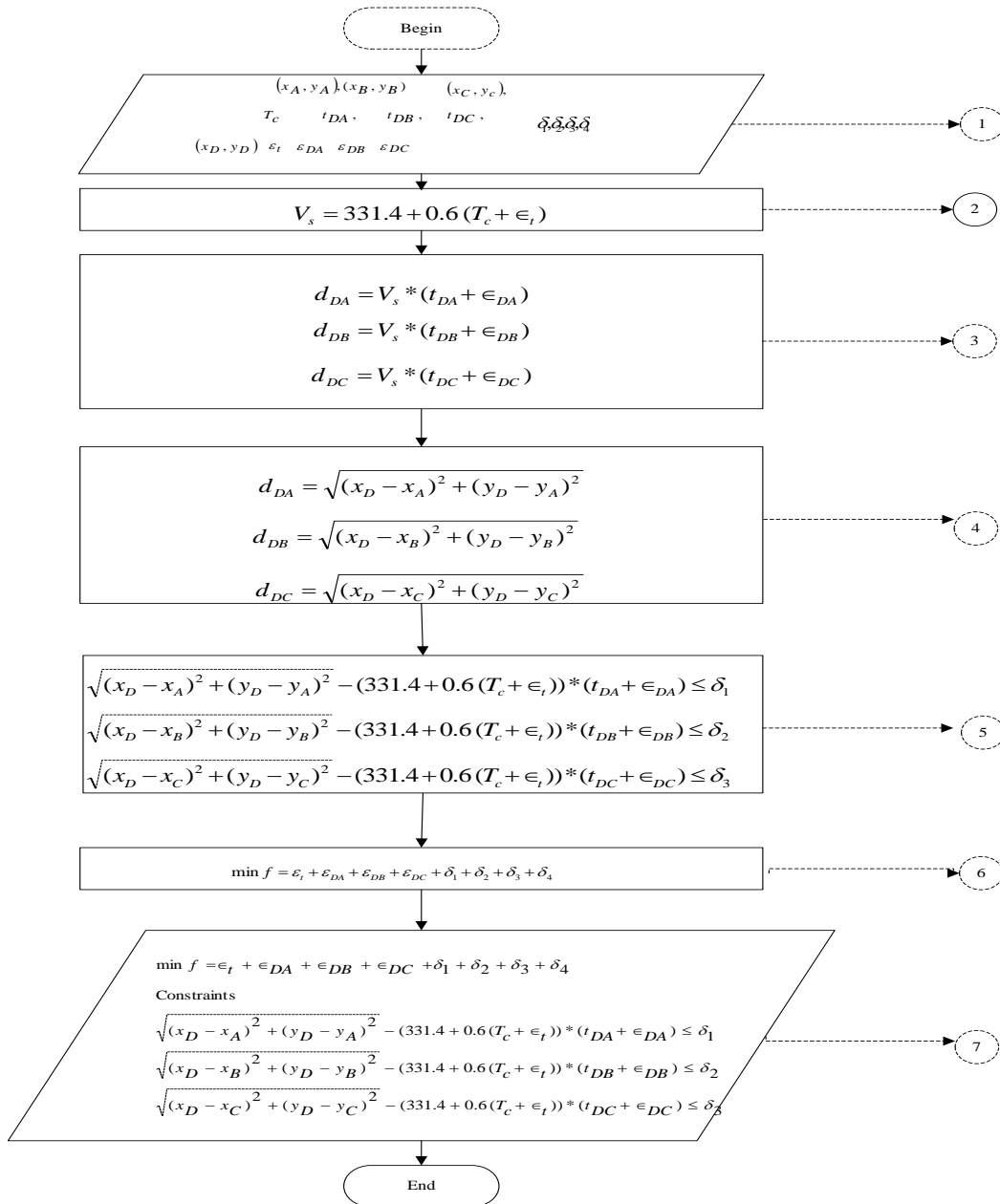


Figure 6.2 Flowchart for generation of cover medium in GPKR watermarking technique

### 6.3.2 Watermark Generation

This subsection explains the process of generating watermark constraints. The requirements for watermark generation process are unique and complex. First, the image is reduced using Gaussian pyramid transforms in order to create the reduced image. Next, the reduced image is converted to a decimal matrix. The decimal matrix is then converted to a binary matrix. The binary matrix contains information that can be used for ownership identification. To generate watermark

constraints, the variables of the binary matrix are numbered using the Kolmogorov rule. These watermark constraints are then embedded into the cover medium. The process of generating watermark consists of three steps:

1. reducing the image using Gaussian pyramid transforms to create a decimal matrix,
2. converting the decimal matrix of the reduced image to a binary matrix, and
3. producing watermark constraints using Kolmogorov rule.


Each of these steps will now be explained in detail.

### 6.3.2.1 Reducing the Image using Gaussian Pyramids Transforms

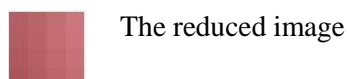
The first step reduces the image captured from the environment by the WMSN nodes to produce the pixels of reduced image. We consider the reduced image in an RGB colour image. The RGB colour image is an  $M \times N \times 3$  array of colour pixels, where each colour is a triplet corresponding to the red, green and blue components of the RGB image. The RGB colour image is of the class of values  $[0,1]$ . The similarity range of the value is  $[0, 255]$  or  $[0, 65535]$  for an RGB image of class uint 8 or uint 16 respectively. The low pixel image can be generated by using the pyramid transforms by implementing the algorithm given below:

Pseudo Code 6.2 Reduction of image by Gaussian Pyramid Transforms

**Input:**

|                                                                                     |                                           |
|-------------------------------------------------------------------------------------|-------------------------------------------|
|  | The sensory image                         |
| N                                                                                   | Number of reducing the image              |
| M                                                                                   | Number is used to bound the reduced image |

**Output :**



**Step 1: Getting all input data**

Prepare the sensory image N, M.

---

**Step 2: Representing the image to the array**

The sensory image is represented by the array  $g_o$  consisting of R rows and C columns of pixels, in order to become the zero level of the Gaussian Pyramids

**Step 3: Getting level 1 of the pyramid transforms**

Low-pass filter the array  $g_o$  to get the level of the pyramid transforms by computing as a weighted average of values in the level 0.

**Step 4: Getting level 2 of the pyramid transforms using the function PYR reduce**

Obtain each value within level 2, representing  $g_2$ , from the values within level 1 by applying the

same pattern of weights. The function PYR reduce is  $g_i(i, j) = \sum_{n=-2}^2 \sum_{m=-2}^2 w(m,n)g_{i-1}(2i+m, 2j+n)$

for level  $0 < l < N$  and nodes  $i, j, 0 \leq i \leq C_i, 0 \leq j \leq R_p$ . N refers to the number of levels in the pyramids, while  $R_p$  and  $C_j$  are the dimensions.

**Step 5: Deciding the number of reduction**

If N, the number of reducing the image, is less than M, go to **step 4**, else go to **Step 6**.

**Step 6: Printing the result of the function PYR reduce**

Represent each value within  $n$  obtained from level  $n-l$  and print the reduced image obtained

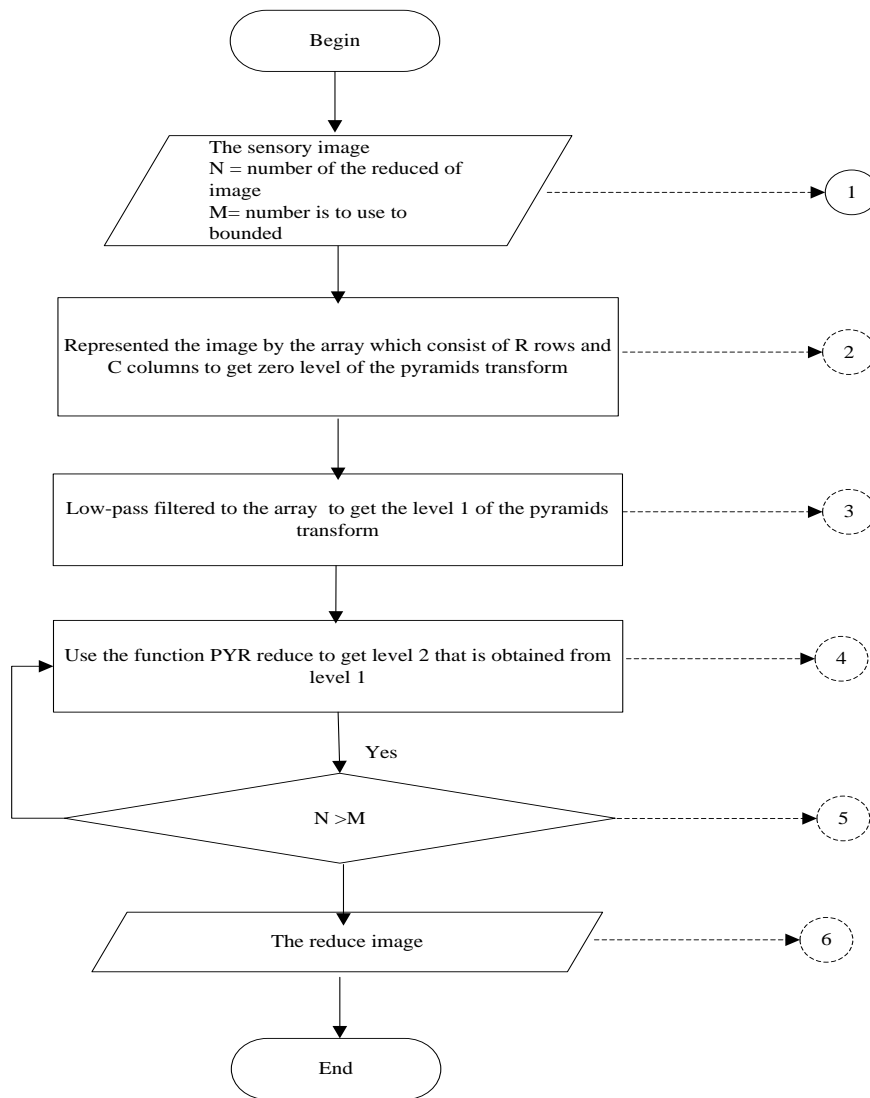


Figure 6.3 Flowchart for getting the reduced image using the Gaussian pyramids

### 6.3.2.2 Converting the Reduced Image to Binary Matrix

The second step is to convert the reduced image to a binary matrix using the function decimal to binary.

Pseudo Code 6.3 Converting the reduced image to binary

**Input:**



The reduced image

**Output**

|   |   |   |   |   |   |   |   |               |
|---|---|---|---|---|---|---|---|---------------|
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | Binary matrix |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |               |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |               |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |               |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |               |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |               |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |               |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |               |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |               |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |               |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |               |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |               |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |               |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |               |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |               |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |               |

**Step 1: Getting all input data**

Obtain the reduced image using Pseudo Code 6.2

**Step 2: Reading the reduced image into a decimal matrix**

Read the array pixel of the reduced image to get the matrix decimal array

**Step 3: Converting the array pixel of the matrix into binary**

Use Pseudo Code 6.4 to convert the decimal matrix into the binary matrix

**Step 4: Printing binary matrix**

Print binary matrix

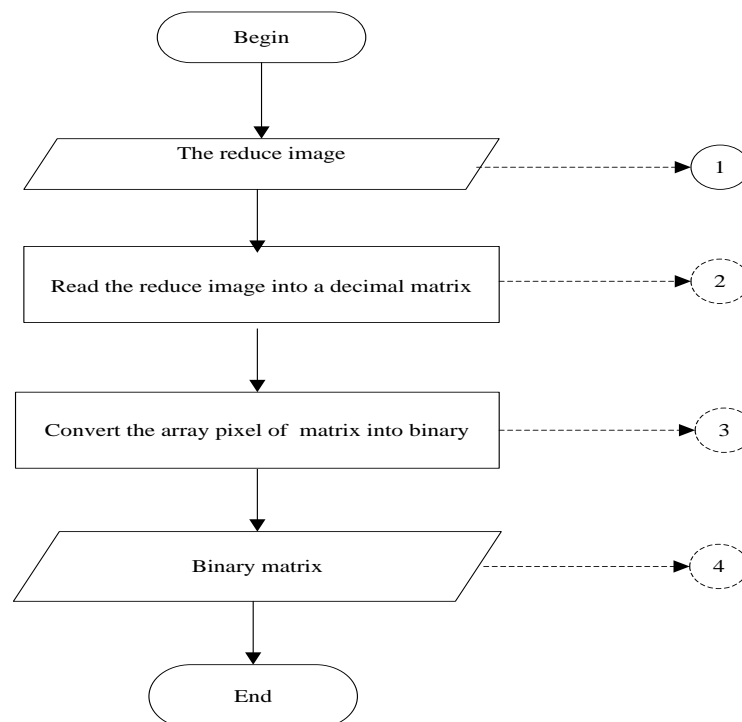


Figure 6.4 Flowchart for converting the reduced image to binary matrix



### 6.3.2.3 Numbering the Watermark Stream using Kolmogorov Complexity Rule

This sub-section explains the creation of watermark constraints using Kolmogorov rules introduced in Section 2.1.2. The rule used here can be seen in Table 6.1.

Table .6.1 The expanded Kolmogorov rule

|              |                 |                 |                 |            |            |            |            |
|--------------|-----------------|-----------------|-----------------|------------|------------|------------|------------|
| 1            | 2               | 3               | 4               | 5          | 6          | 7          | 8          |
| $\epsilon_t$ | $\epsilon_{DA}$ | $\epsilon_{DB}$ | $\epsilon_{DC}$ | $\delta_1$ | $\delta_2$ | $\delta_3$ | $\delta_4$ |

The binary matrix produced using the pseudo code is numbered using the Kolmogorov rule given in Table 6.1

If a bit one is assigned to a variable within a group, that variable is included in the linear of variables from the group. If a bit zero is assigned to a variable within a group, that variable is not included. According to this rule, we add some new constraints into the linear combination of variables.

The pseudo-code for generating watermark constraints is shown in Pseudo Code 6.3 and the flowchart in Figure 6.5.

Pseudo Code 6.4 Generating watermark constraints

**Input:**

```

1 0 1 0 1 1 0 1
1 1 0 0 1 1 0 1
0 0 0 0 1 1 0 1
0 0 1 1 0 1 0 1
1 1 0 1 1 1 0 1
1 0 0 1 1 1 0 1
0 1 1 0 1 1 0 1
1 1 0 0 1 1 0 1
0 0 1 0 0 0 1 1
1 1 0 0 0 0 1 1
0 0 0 0 0 0 1 1
0 0 1 1 1 1 0 1
1 0 1 1 0 0 1 1
1 1 0 1 0 0 1 1
0 0 0 1 0 0 1 1
0 0 1 0 0 0 1 1

```

Binary matrix

**Output**

Watermark constrains

```

d1 + d2 + d4 + edb + et
d1 + d2 + d4 + eda + et
 d1 + d2 + d4
 d2 + d4 + edb + edc
d1 + d2 + d4 + eda + edc + et
 d1 + d2 + d4 + edc + et
 d1 + d2 + d4 + eda + edb
 d1 + d2 + d4 + eda + et
 d3 + d4 + edb
 d3 + d4 + eda + et
 d3 + d4
d1 + d2 + d4 + edb + edc
d3 + d4 + edb + edc + et
d3 + d4 + eda + edc + et
 d3 + d4 + edc
 d3 + d4 + edb

```

**Step 1 : Getting input data**

Read binary matrix.

**Step 2: Matching, using table 6.1**

Match the bit number with the corresponding variable number using Kolmogorov rule (Table 6.1 ).

**Step 3: Deciding whether a variable is included or not**

If a bit one is assigned to a variable within a group, include that variable in the linear and go to step 4. Else, if a bit zero is assigned to a variable within a group, do not include that variable in the linear and go to step 2.

**Step 4: Generating watermark constraints**

Print WC (watermark constraints).

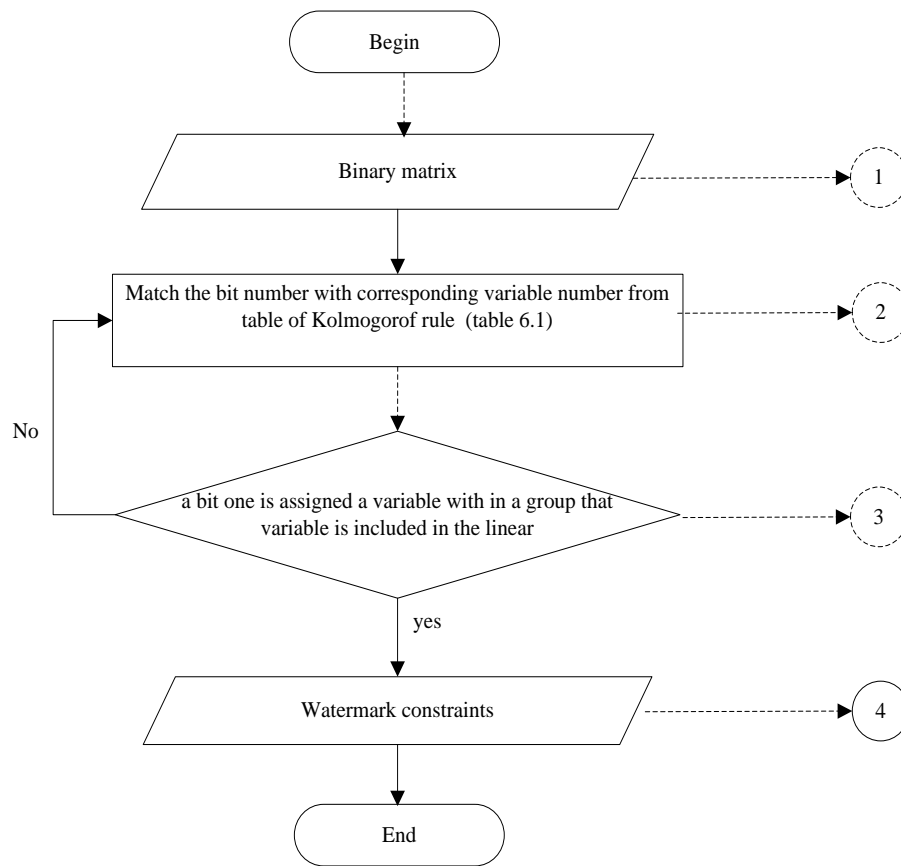


Figure 6.5 Flowchart for generating watermark constraints

### 6.3.3 Watermark Embedding Algorithm

This sub-section explains the process of embedding watermark constraints. This is the second step of the watermarking system and is undertaken by an embedder. The embedder combines the cover medium and creates a watermarked cover medium. The watermarked cover medium is perceptibly identical to the cover medium.

Pseudo Code 6.5 Embedding watermark constraints

**Input**

|                                      |                                             |
|--------------------------------------|---------------------------------------------|
| WC                                   | Watermark constraints                       |
| $(x_A, y_A), (x_B, y_B), (x_C, y_C)$ | Position of two-dimensional sensor networks |
| $T_c$                                | Temperature of the propagation media        |

|                                  |                                                                                                                   |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------|
| $\tau_1, \tau_2, \tau_3, \tau_4$ | The values are selected such that the feasibility of the solution space of the optimization problem is not harmed |
| $t_{DA}, t_{DB}, t_{DC}$         | Time transmission between node D to A, D to B, and D to C                                                         |

**Output**

|                                                        |                                                                                                                                           |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| $(x_D, y_D)$                                           | Position of two-dimensional sensor networks                                                                                               |
| $\delta_1, \delta_2, \delta_3$                         | Error in the measurement between the Euclidean measurement and that measured using time differences of optimal D to A, D to B, and D to C |
| $\varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}$ | Error in the measurement of the timer from D to A, D to B, and D to C                                                                     |
| $\delta_4$                                             | Auxiliary variable                                                                                                                        |
| $\varepsilon_t$                                        | Error in the measurement of temperature                                                                                                   |
| $min f$                                                | Minimum function of the objective                                                                                                         |

**Step 1 : Getting all input Data**

Obtain the three positions  $(x_A, y_A), (x_B, y_B), (x_C, y_C)$  randomly from the network of 75 positions of two dimensional sensor nodes. Obtain the temperature of the propagation media ( $T_c$ ) by using uniform distribution on interval [0,50]. Obtain time transmission of  $t_{DA}, t_{DB}$ , and  $t_{DC}$  by using Gauss distribution on interval [0, 1]. Obtain the values of  $\tau_1, \tau_2, \tau_3$ , and  $\tau_4$  using Gauss distribution on interval [0, 1]. Watermark constraints (WC) are obtained by applying Pseudo Code 6.4.

**Step 2: Using the cover medium**

Obtain cover medium by applying Pseudo Code 6.1, called NLSP, with the objective function consisting of the coefficient objective.

**Step 3: Appending watermark constraints (WC)**

Append watermark constraints to the cover medium.

**Step 4: Computing new cover medium to solve**

Add the message sensed data and watermark constraints to compute a new cover medium, use TOMLAB to solve it, and then obtain  $(x_D, y_D), \varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4$  and  $min f$ .

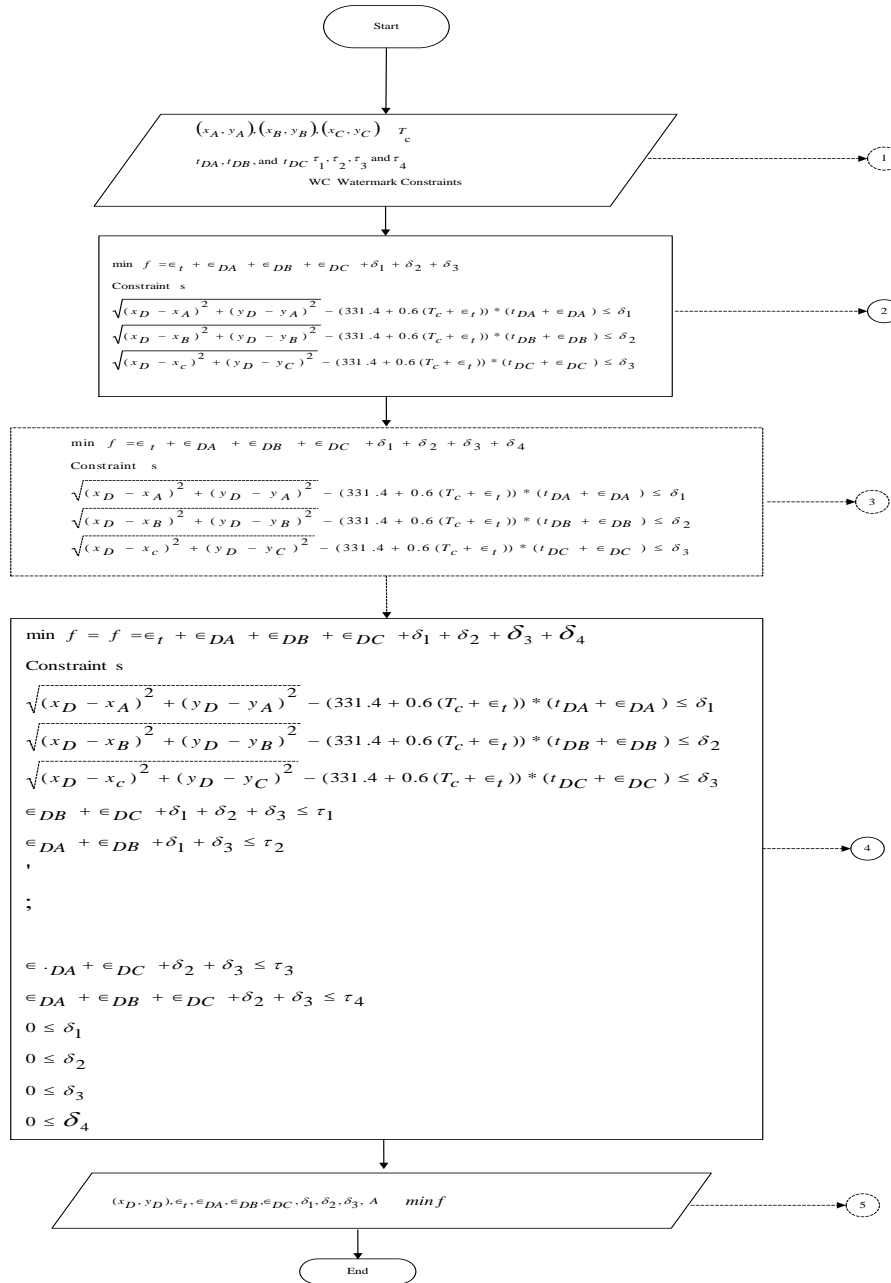


Figure 6.6 Watermark constraints and digital message image embedding process

### 6.3.4 The Process of Extraction and Detection of the Watermark

This sub-section explains the process of extracting and detecting the watermark which is undertaken by a detector. The extraction process can be divided into two phases: locating the watermark and recovering the watermark information. There are two types of detection: informed

detection and blind detection depending on whether the cover medium is required or not in the detection process. The GPKR watermarking technique uses blind detection to detect watermark.


### 6.3.4.1 Watermark Extraction

The extraction process is also undertaken in watermark detector as the reduced image needs to be expanded. The extraction process uses a function `PYR_EXPAND`. This function is the reverse of `PYR_REDUCE`. This function expands an  $(M + 1)$ -by- $(N + 1)$  array into a  $(2M + 1)$ -by- $(2N + 1)$  array by interpolating new node values. So `PYR_EXPAND` applied to array  $g_1$  of the Gaussian pyramid would result in an array  $g_{l,1}$  which is of the same size as  $g_{l-1}$ .


The process of extracting the image can be depicted in Pseudo Code 6.6

Pseudo Code 6.6 Expanding the reduced image by the pyramid transforms to get the sensory image

**Input:**

|                                                                                                                                                       |                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| $(x_D, y_D), \epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4$<br><br>$\tau_1, \tau_2, \tau_3, \tau_4$ | All of the errors of measurement |
| <b>WC</b>                                                                                                                                             | Watermark constraints            |
|                                                                    | The reduced image                |

**Output**

|                                                                                     |                   |
|-------------------------------------------------------------------------------------|-------------------|
|  | The sensory image |
|-------------------------------------------------------------------------------------|-------------------|

**Step 1: Getting all input data**

Get all of the errors of measurement  $(x_D, y_D), \epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4, \tau_1, \tau_2, \tau_3, \tau_4$ ,

WC (watermark constraints) and the reduced image.

**Step 2: Detecting all of the errors of measurement**

Use  $(x_D, y_D), \epsilon_i, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4$  and  $\tau_1, \tau_2, \tau_3, \tau_4$  to detect whether the watermark constraints satisfy all the requirements. If they satisfy all the errors of measurement, go to **Step 3**, otherwise go to **Step 1**.

### **Step 3: Converting coefficient watermark constraints into binary matrix**

Convert coefficient watermark constraints into binary matrix using the inverse Kolmogorov rule, and then convert the binary matrix to decimal matrix.

### **Step 4: Expanding the reduced image using the function PYR\_Expand to get the original image**

Apply array  $g_1$  of the Gaussian pyramid to the reduced image, in order to get the array  $g_{l,1}$  which is of the same size as  $g_{l-1}$ .

Let  $g_{l-n}$  be the result of expanding  $g_l$  n times.

$g_{l,0} = g_1$  and  $g_{l,n} = \text{PYR\_EXPAND}(g_l, n-1)$ . PYR\_REDUCE means for level  $0 < l < N$ ,  $0 \leq N$  and

$$\text{nodes } i, j, 0 \leq i \leq C_i, 0 \leq j \leq R_p \quad g_{l,n}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l,n-1} \left( \frac{i-m}{2}, \frac{j-n}{2} \right)$$

### **Step 5: Applying function PYR\_Expand to get the same size as the original image**

Apply the function PYR\_EXPAND n times to the image  $g_l$  to obtain  $g_{l,l}$  which is of the same size as the original image.

### **Step 6: Printing the sensory image**

Print the sensory image.

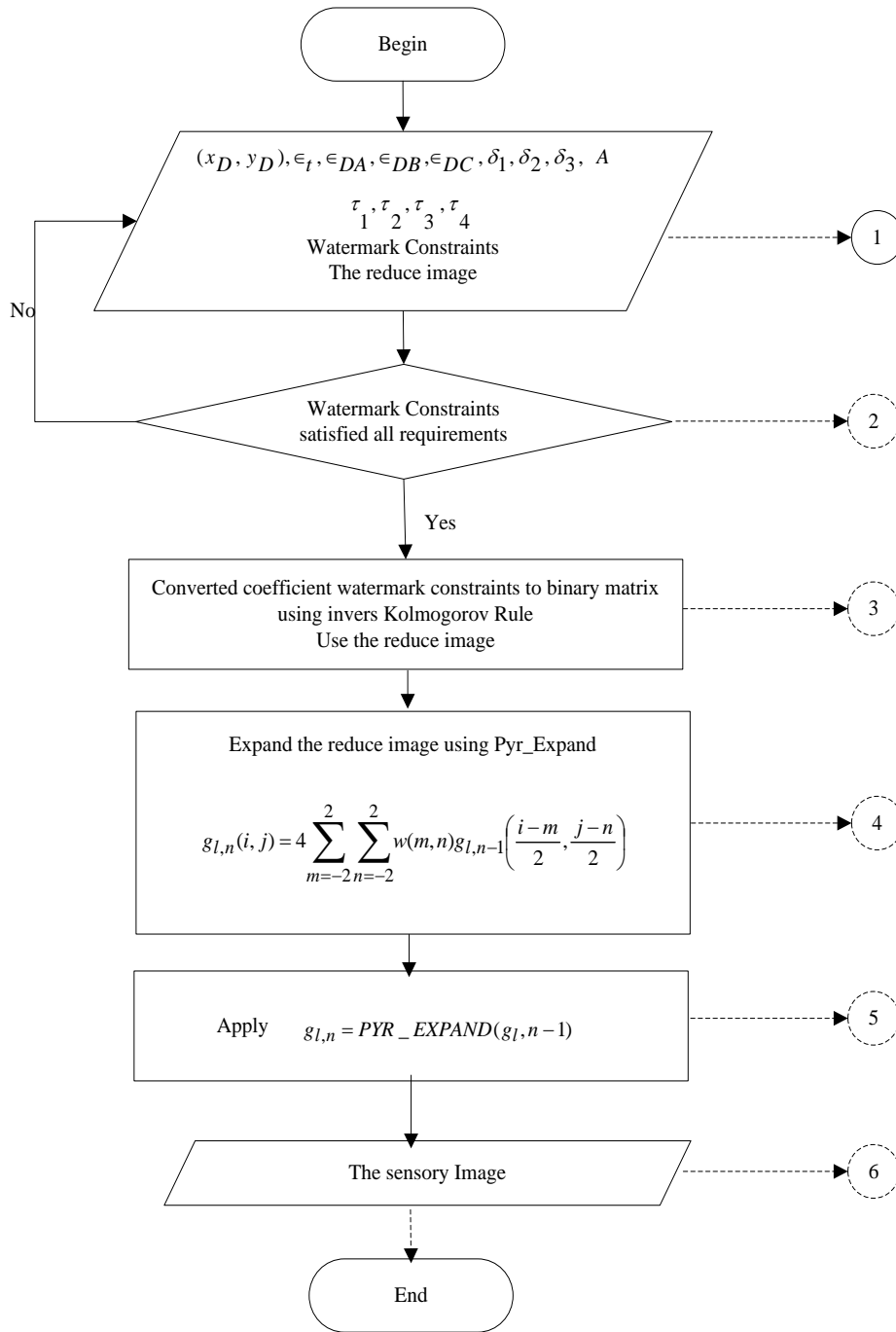


Figure 6.7 Flowchart for extraction process

### 6.3.4.2 Watermark Detecting Process

This sub-section explains the detection process. There are two types of detection: informed detection and blind detection, depending on whether the cover medium is required or not in the detection process. To verify the presence of the watermark constraints, we adopt the concept of



Cox *et al.* (Cox, Kilian, and Leighton 1997). Let  $x$  be the error of the optimal solution without watermark constraints,  $x'$  the error of the optimal solution with watermark constraints, and  $x''$  the error of the optimal solution with watermark constraint attacks. For detecting the watermark, a correlation value or similarity measure is used in most of these methods. Here, to verify the presence of the watermark constraints, the difference measure between the normalized difference errors of the optimal solution without watermark constraints and with watermark constraints

( $C = x' - x$ ) is obtained. The similarity measure between the normalized difference errors of the optimal solution without watermark constraints and with watermark constraint attacks is ( $C' = x'' - x'$ ). The similarity measure is given by the normalized correlation coefficient

$$SM(C', X') = \frac{C' \cdot X'}{\sqrt{X' \cdot X'}}$$

The pseudo-code for detecting watermark signal is shown in Pseudo Code 6.7 and the process of watermark detection is shown in Figure 6.8.

Pseudo Code 6.7 The process of detecting watermark constraints

**Input**

| WC                                                                                                                        | Watermark constraints                                       |
|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| $x = [\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4]$                   | Error of the optimal solution without watermark constraints |
| $x' = [\epsilon'_t, \epsilon'_{DA}, \epsilon'_{DB}, \epsilon'_{DC}, \delta'_1, \delta'_2, \delta'_3, \delta'_4]$          | Error of the optimal solution with watermark constraints    |
| $x'' = [\epsilon''_t, \epsilon''_{DA}, \epsilon''_{DB}, \epsilon''_{DC}, \delta''_1, \delta''_2, \delta''_3, \delta''_4]$ | Error of the optimal watermark constraint attacks           |

**Output**

Detecting whether the watermark constraints (WC) are robust or not

**Step 1: Getting all input data**

$$x = [\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}, \delta_1, \delta_2, \delta_3, \delta_4], \quad x' = [\epsilon'_t, \epsilon'_{DA}, \epsilon'_{DB}, \epsilon'_{DC}, \delta'_1, \delta'_2, \delta'_3, \delta'_4],$$

$$x'' = [\epsilon''_t, \epsilon''_{DA}, \epsilon''_{DB}, \epsilon''_{DC}, \delta''_1, \delta''_2, \delta''_3, \delta''_4] \text{ and Watermark Constraints.}$$

**Step 2: Computing the difference errors of the optimal solution with watermark constraints, computing threshold and similarity.**

Compute  $c = x' - x$  and  $c'' = x'' - x$ ,  $\text{threshold} = \frac{cx'}{\sqrt{x'x'}}$  and  $\text{similarity} = \frac{cx''}{\sqrt{x''x''}}$ .

**Step 3: Deciding whether watermark constraints are robust**

If  $\text{threshold} \geq \text{similarity}$ , go to watermark constraint is robust.

**Step 4: Deciding whether watermark constraints are not robust**

If  $\text{threshold} < \text{similarity}$ , go to watermark constraint is not robust.

---

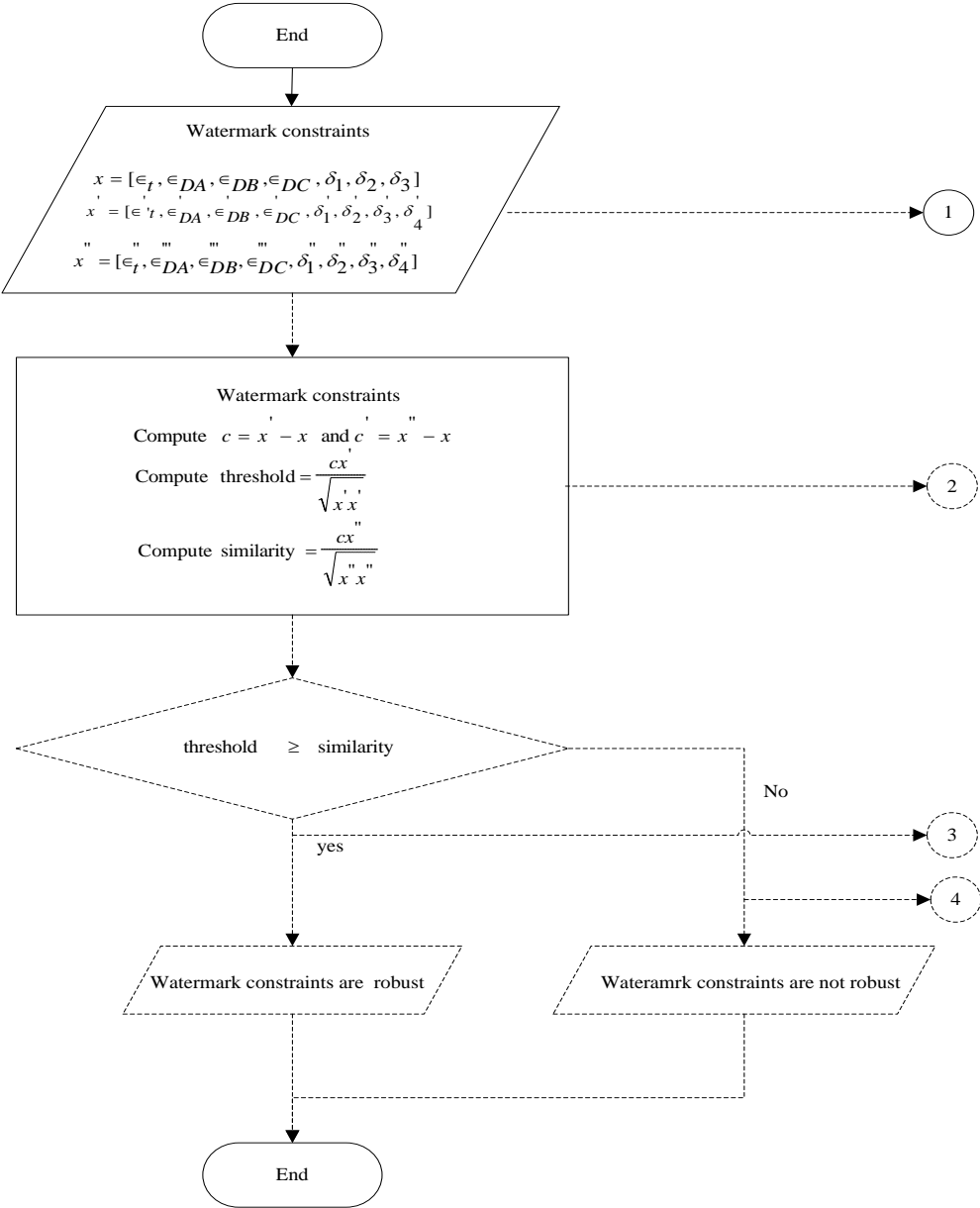


Figure 6.8 Flow chart for the detection process of the GPKR watermarking technique

### 6.4 Implementation of the Prototype

This section describes the experimental set-up used for an extensive testing of the model, to check whether the proposed algorithm is feasible. MATLAB and TOMLAB were used in the experimental set-up. This represents the fourth stage of the conceptual process described in Section 4.4, in which the theoretical foundation was implemented as a prototype. We modelled network set-ups for the processes of generating watermark, embedding watermark constraints, and

extracting watermark signal, in the overall set-up for copyright protection of images in WMSNs. We also modelled some attacks to this copyright protection model often used by attackers, such as modifying the different results of the embedding process. The overall process of copyright protection using watermark was modelled using MATLAB files as given below:

*MenuGPKR.m*

*NetworkSetupWMSNs.m*

*GenerateCoverMedium.m*

*GenerateWatermark.m (pyr\_reduce.m, Kolmogov\_rule\_WMSN)*

*ComputationCoverMedium.m*

*EmbeddingWithCONSTRAINTS.m*

*ExtractWMSN.m*

*DetectingProcessWMSNs.m*

*DetecAttackDelete.m*

*DetectAttackFalseInsert.m*

*DetectAttackModification.m*

*DetectAttackReplication.m*

The operations were governed by *menGPKR.m* which initialized all the required variables, such as the path to network set-up, cover medium set-up and external applications, and passed the embedded and extracted parameters to the file *Extract.m* for processing. Attacks were performed within the file *menuGPKR.m* and are described in Section 6.4.5. For verifying whether the watermark signal was present, the similarity between the normalized difference error from the optimal solution between the watermarked solution  $X'$  and the solution obtained without watermark  $X$  was applied, and was modelled using the file *DetectingRobustWateramrk.m*

---

### 6.4.1 Source Code : Network Set -Up Generation

The process of network setting was modelled using the file *NetworkSetupWMS.m*. The call to *NetworkSetupWMSNs.m* from *menuGPKR.m* passed to the location of the network setting. The pseudo code for the network setting process and further details have been given in Section 6.5.1.

### 6.4.2 Source Code : Cover Medium Generation

The process of cover medium generation was modelled using the file *GenerateCoverMedium.m*. The call to *GenerateCoverMedium.m* from *menuGPKR.m* passed to the location of the cover medium and the parameters for the cover medium to perform the embedding operation. The pseudo code for the cover medium generation has been explained in Section 6.5.1.

Figure 6.9 shows the screenshot of cover medium generation using MATLAB Code.

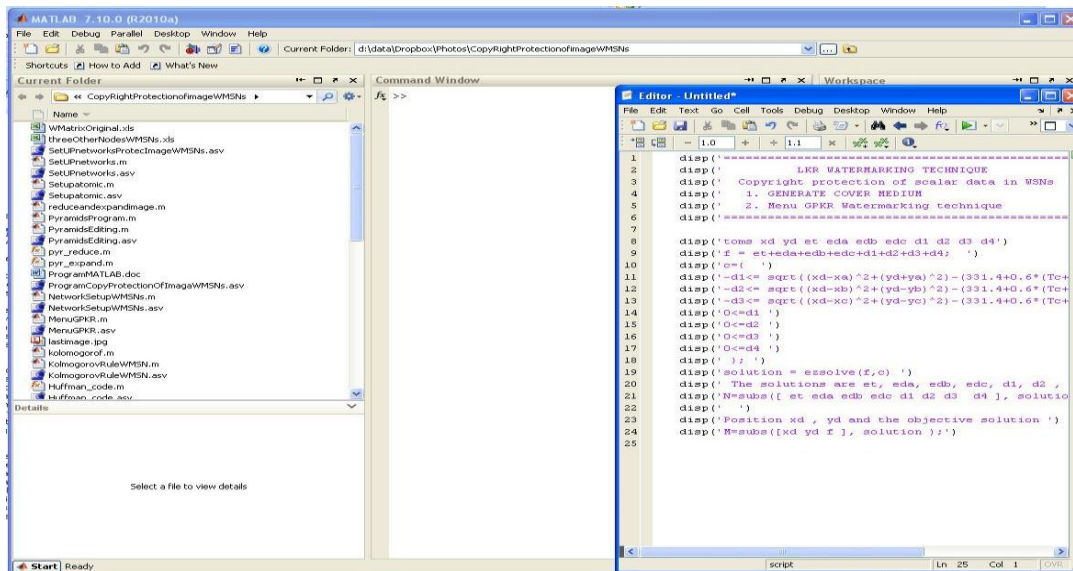


Figure 6.9 Screenshot of cover medium generation using MATLAB Code

### 6.4.3 Source Code : Watermark Generation

The process of watermark generation consists of four steps: the process of reducing image was modelled using the MATLAB function *pyr\_reduce.m*, the process of converting the RGB colour image to decimal matrix signals was modelled using the file *de2bi.m* and the process of producing watermark constraints was modelled using the file *GenerateWatermark.m* (*pyr\_reduce.m*,

Kolmogov\_rule\_WMSN). The pseudo code for watermark generation has been given in Section 6.3.2. The screenshot for cover medium generation using MATLAB Code has been shown in Figure 6.10.

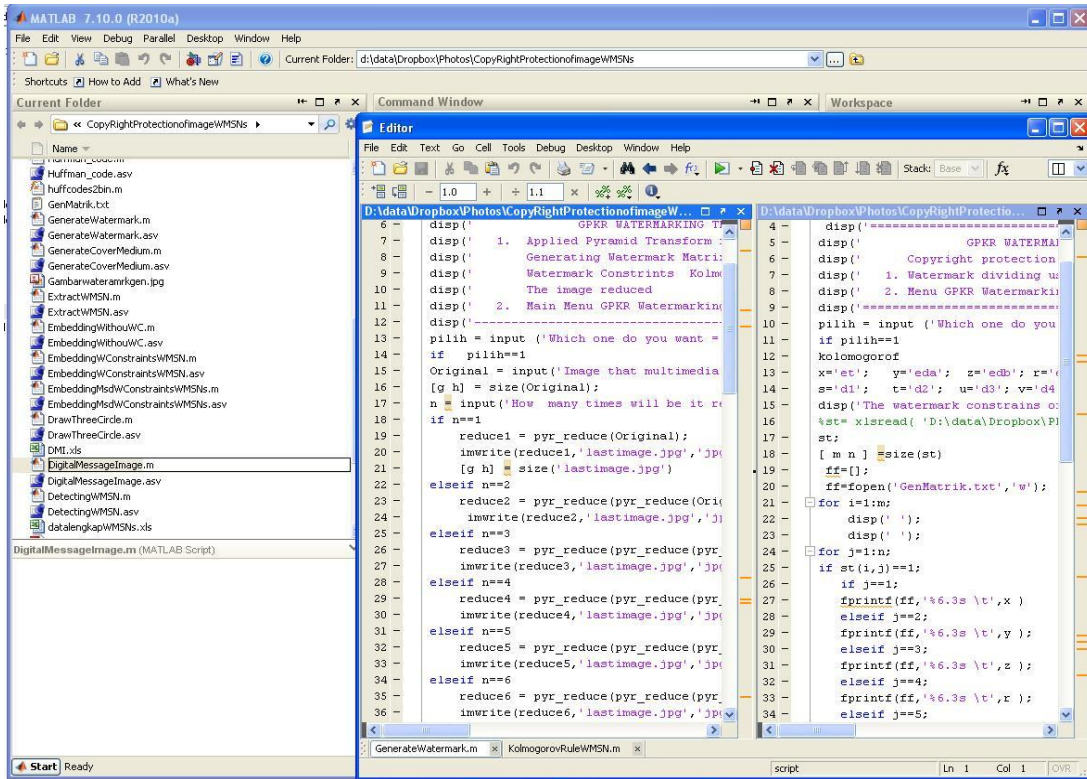


Figure 6.10 Screenshot of cover medium generation using MATLAB Code

## 6.4.4 Source Code : Embedding Watermark

The process of embedding watermark constraints was modelled using the file *EmbeddingWithCONSTRAINTS.m*. The call to *EmbeddingWithCONSTRAINTS* from *menuGPKR.m* passed to the location of the cover medium and the parameters to perform the embedding operation. The pseudo code for the process of embedding watermark constraints has been described in Section 6.3.3.

Figure 6.11 shows the screenshot of the process of embedding watermark constraints.

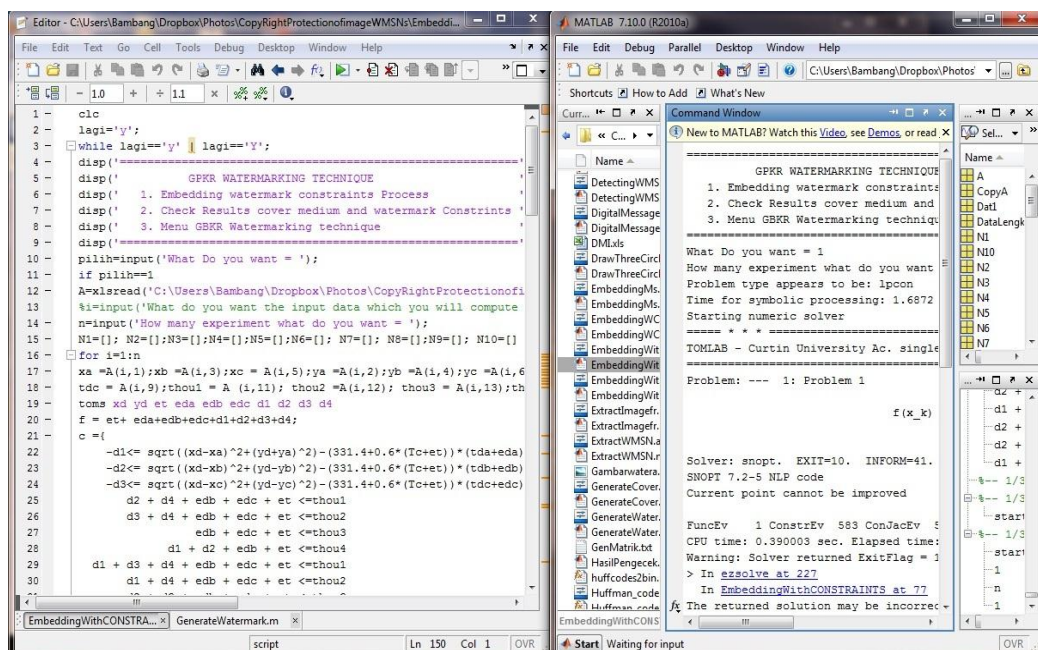


Figure 6.11 Screenshot of the process of embedding watermark constraints

## 6.4.5 Source Code : Extracting and Detecting

The process of extracting the watermark signal was modelled using the file *ExtractWMSN.m*. The pseudo code for the process of extracting the watermark signal has been described in section 6.3.4.

Figure 6.12 shows the screen shot of the process of extracting the watermark signal





### 6.4.5.1 Source Code: Attacks

We assume that the results of the process of embedding watermark constraints are estimated by an attacker, and can be removed, manipulated, modified or changed. The corresponding attacks on the embedding process are:

1. Deleting a number of the results of the embedding process in the hope that the receiver will not get an appropriate expanded image. The process of deleting a number of results was modelled by the file *AttackDELETEimageWMSN.m*.
2. Inserting false results of the embedding process in the hope that the new results of the watermark constraints embedding process will map into the results of the watermark embedding process. The process of inserting false results was modelled by the file *DetectAttackFalseInsert.m*.
3. Modifying the results of the watermark embedding process in the hope to find new results of watermark embedding process that will map into the results of the watermark embedding process. The process of modifying the results was modelled by the file *DetectAttackModification.m*.
4. Replicating different results of the watermark constraints embedding process in the hope to find new results of error of the watermark constraints embedding process that will map into the existing solution. This process of replicating watermark constraints was modelled using the file *DetectAttackReplication.m*.

All the implementation details of different types of attack, viz. deletion, false insertion, modification and replication, have been given in Section 6.5.3.

## 6.5 Experimental Setting

This section describes the experimental set-up used for an extensive testing of the model. In this case, we did not use a square shaped area for deploying the multimedia sensor nodes. Instead, we deployed the multimedia sensor nodes over a rectangular shaped area. One reason for using a

---

rectangular area was to find out differences, if any, from the use of square meter. We randomly placed 50 nodes within a 200 meter length and 100 meter width, as shown in Figure 6.16.

### 6.5.1 Network Set-Up

This section describes the process of setting the network for testing our model of copyright protection of images in WMSNs. We generated 32 positions of  $(x_A, y_A), (x_B, y_B)$  and  $(x_C, y_C)$  using random positions of 50 multimedia sensor nodes. We further generated  $\tau_c$  using Gauss distribution on interval  $[0, 1]$ , 32  $t_{DA}, t_{DB}$  and  $t_{DC}$  using uniform distribution on interval  $[0, 1]$ , and  $\tau_1, \tau_2, \tau_3$  and  $\tau_4$  using Gauss distribution on interval  $[0, 1]$ , so that these values do not harm the feasibility of solution of the cover medium.

The pseudo code for setting the network is shown in Pseudo Code 6.8.

Pseudo Code 6.8 Network set-up for GPKR watermarking technique

#### Input

|   |                                                                 |
|---|-----------------------------------------------------------------|
| N | Number of Multimedia sensor nodes                               |
| L | Unit length of multimedia sensor nodes                          |
| M | Unit width of multimedia sensor nodes                           |
| R | Maximum range of two sensor nodes communicating with each other |

#### Output

|                                        |                                                                                                            |
|----------------------------------------|------------------------------------------------------------------------------------------------------------|
| $(x_i, y_i), i=1, \dots, 50$           | Coordinate positions of two-dimensional sensor nodes                                                       |
| $\tau_c$                               | Temperature of the propagation media                                                                       |
| $t_{DA}, t_{DB}$ and $t_{DC}$          | Generation of time transmission between two sensor nodes                                                   |
| $\delta_1, \delta_2$ and $\delta_3$    | Errors between the Euclidean distances measured from the feasibility of the solution space of optimization |
| $\delta_4$                             | Auxiliary variables to the cover medium                                                                    |
| $\tau_1, \tau_2, \tau_3,$ and $\tau_4$ | Value of the feasibility of the solution space of optimization                                             |

#### Steps 1: Setting all data

$N = 50, L = 200, M = 100$  and  $R = 30$

---

Set 50 multimedia sensor nodes randomly within a 200 meter length and 100 meter width, and set maximum range of two sensor nodes communicating with each other at  $R=30$ .

**Step 2: Positioning and plotting the coordinates**

Generate the coordinates of 50 multimedia sensor nodes' position of  $x$ ,  $y$ , and plot these coordinates.

**Step 3: Plotting a link between two sensor nodes using Euclidean Theorem**

Compute the distance between two multimedia sensor nodes using Euclidean Theorem. If distance is less than  $R$ , there is a link between the two multimedia sensor nodes, else there is no link between the two multimedia sensor nodes.

**Step 4: Generating parameters of time transmission, temperature, errors between two Euclidean distances, and the value of feasibility of the solution space**

Generate time transmission between two sensor nodes based on uniform distribution over the interval  $[0, 1]$ , Generate the temperature of the propagation media based on random distribution over the interval  $[1, 50]$ , Generate errors between the Euclidean distances measured based on uniform distribution over the interval  $[0, 1]$  and the feasibility of the solution space of optimization based on uniform distribution over the interval  $[0,1]$ .

**Step 5: Printing coordinate positions, temperature, time transmission, error between two distances measured, and the value of the visibility**

Print coordinate positions  $(x_i, y_i), i = 1, 2, 3, \dots, 50$ , temperature  $T_c$ , time transmission  $t_{DA}, t_{DB}, t_{DC}$ , and the value of visibility  $\tau_1, \tau_2, \tau_3$ , and  $\tau_4$

Figure 6.14 Shows the flowchart for network set-up of the GPKR watermarking technique

---

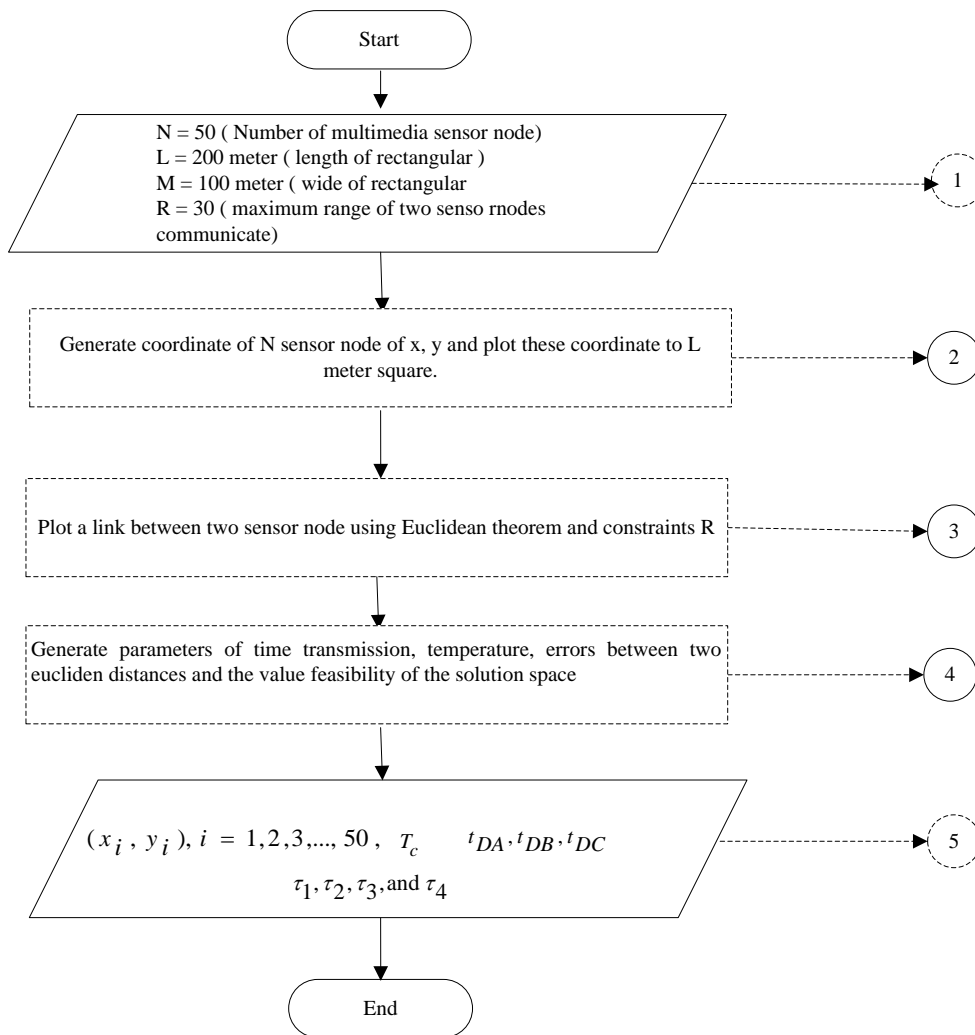


Figure 6.14 Flowchart for network set-up of GPKR watermarking technique

We implemented the network set-up for the GPKR watermarking technique through Pseudo Code 5.6, using MATLAB given in Figure 6.15. The network setting is illustrated in Figure 6.16.

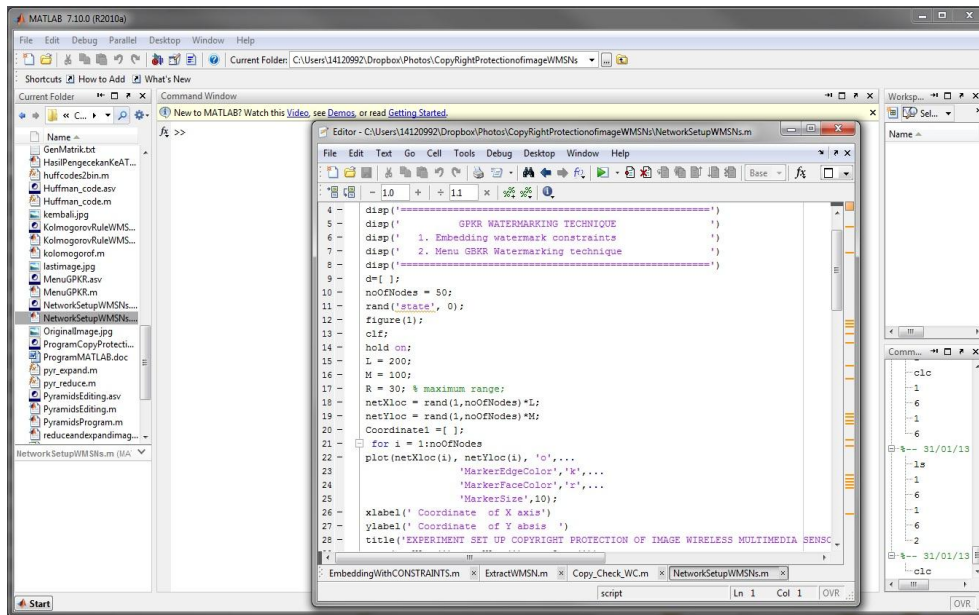


Figure 6.15 Screenshot of network setting for GPKR watermarking technique using MATLAB Code

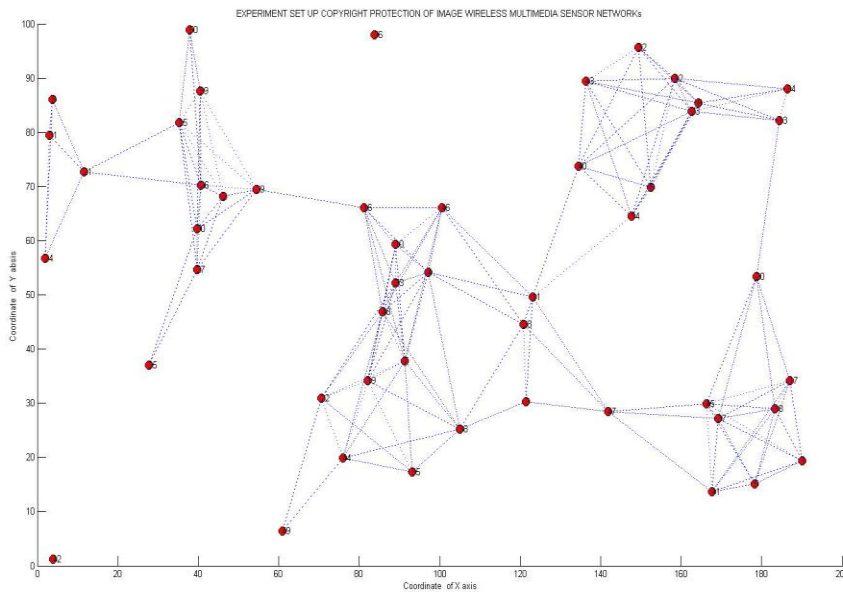


Figure 6.16 50 nodes randomly deployed within a 200 meter length and 100 meter width  
 ased on Figure 6.16, we can list the coordinates of 50 nodes, as shown in Table 6.2.

Table 6.2 Coordinate positions of 50 nodes

| No | X        | Y        | No | X        | Y        |
|----|----------|----------|----|----------|----------|
| 1  | 190.0259 | 19.34312 | 26 | 40.55304 | 70.27399 |
| 2  | 46.2277  | 68.22232 | 27 | 39.74435 | 54.65712 |
| 3  | 121.3685 | 30.27644 | 28 | 120.7585 | 44.48802 |
| 4  | 97.19649 | 54.16739 | 29 | 54.43758 | 69.45672 |
| 5  | 178.2598 | 15.0873  | 30 | 39.76285 | 62.13101 |
| 6  | 152.4194 | 69.78985 | 31 | 3.054785 | 79.48211 |
| 7  | 91.29353 | 37.8373  | 32 | 149.3571 | 95.68434 |
| 8  | 3.700729 | 86.00116 | 33 | 89.01929 | 52.25903 |
| 9  | 164.2814 | 85.36551 | 34 | 186.3629 | 88.01422 |
| 10 | 88.94067 | 59.35629 | 35 | 93.19887 | 17.29561 |
| 11 | 123.0865 | 49.65524 | 36 | 83.72989 | 97.97469 |
| 12 | 158.3874 | 89.97692 | 37 | 169.2443 | 27.14473 |
| 13 | 184.3626 | 82.16292 | 38 | 105.0305 | 25.23293 |
| 14 | 147.6414 | 64.49104 | 39 | 40.52947 | 87.57419 |
| 15 | 35.25323 | 81.79743 | 40 | 134.4275 | 73.7306  |
| 16 | 81.14124 | 66.02276 | 41 | 167.6237 | 13.65187 |
| 17 | 187.0939 | 34.19706 | 42 | 3.927903 | 1.175669 |
| 18 | 183.3809 | 28.97259 | 43 | 136.2554 | 89.3898  |
| 19 | 82.05404 | 34.11936 | 44 | 75.8962  | 19.91381 |
| 20 | 178.7299 | 53.4079  | 45 | 166.3592 | 29.8723  |
| 21 | 11.57826 | 72.71132 | 46 | 100.5626 | 66.14426 |
| 22 | 70.57363 | 30.92902 | 47 | 141.8943 | 28.44086 |
| 23 | 162.6333 | 83.8496  | 48 | 85.77847 | 46.92243 |
| 24 | 1.97226  | 56.80725 | 49 | 60.92347 | 6.478112 |
| 25 | 27.77818 | 37.04136 | 50 | 37.93075 | 98.83349 |

The next step of experimental setting is the generation of cover medium algorithm by using Pseudo Code 6.8, which was explained in Section 2.2.1. We obtain the NLSP as follows:

$$\min f = \epsilon_t + \epsilon_{DA} + \epsilon_{DB} + \epsilon_{DC} + \delta_1 + \delta_2 + \delta_3 + \delta_4$$

Constraints

$$\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DA} + \epsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DB} + \epsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_c)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \epsilon_t)) * (t_{DC} + \epsilon_{DC}) \leq \delta_3$$

$$0 \leq \delta_1$$

$$0 \leq \delta_2$$

$$0 \leq \delta_3$$

$$0 \leq \delta_4$$

We then provide all the data input which is used for preparing the NLSP. To provide all the data we:

- 
1. generate 32 of  $(x_A, y_A), (x_B, y_B)$  and  $(x_C, y_C)$  using random positions of 50 sensor nodes within 200 meter length and 100 meter width.
  2. generate 32  $t_{DA}, t_{DB}$  and  $t_{DC}$  using uniform distribution over the interval  $[0, 1]$ .
  3. generate 32  $\tau_c$  using Gauss distribution on interval  $[0, 1]$ .
  4. generate 32  $\tau_1, \tau_2, \tau_3$  and  $\tau_4$  using Gauss distribution over the interval  $[0, 1]$ , so that these values do not harm the feasibility of the solution of cover medium .

Table 6.3 shows all the data input for our model of copyright protection of images in WMSNs

---

Table 6.3 32 experiments of the positions of  $(x_A, y_A)$ ,  $(x_B, y_B)$ ,  $(x_C, y_C)$  and 32 experiments time measurement by using Gaussian distribution [0,1], 32 experiment of temperature random between [0,50] and the values of  $\tau_1, \tau_2, \tau_3$ , and  $\tau_4$  generated by using normal distribution [0,1].

| No | Position of three sensor nodes |          |          |          |          |          |          |          |          | The exact time |          |          |          | temperature | The feasibility of value |       |  |  |
|----|--------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------------|----------|----------|----------|-------------|--------------------------|-------|--|--|
|    | Xa                             | Ya       | Xb       | Yb       | Xc       | Yc       | Tda      | Tdb      | Tdc      | Td             | Tc       | thou1    | thou2    |             | thou3                    | thou4 |  |  |
| 1  | 190.0259                       | 19.34312 | 97.19649 | 54.16739 | 91.29353 | 37.8373  | 0.118738 | 0.845702 | 0.155561 | 8.354350415    | 0.181278 | 0.796183 | 0.737905 | 0.715248    |                          |       |  |  |
| 2  | 46.2277                        | 68.22232 | 178.2508 | 15.0873  | 3.700729 | 86.00116 | 0.5077   | 0.838624 | 0.913642 | 8.79031964     | 0.494466 | 0.083359 | 0.886622 | 0.271568    |                          |       |  |  |
| 3  | 121.3685                       | 30.27644 | 152.4194 | 69.78985 | 164.2814 | 85.36551 | 0.520481 | 0.20552  | 0.992633 | 23.2764426     | 0.126418 | 0.934844 | 0.289504 | 0.109368    |                          |       |  |  |
| 4  | 97.19649                       | 54.16739 | 91.29353 | 37.8373  | 88.94067 | 59.35629 | 0.71278  | 0.412435 | 0.828656 | 13.29858649    | 0.383192 | 0.166776 | 0.735167 | 0.127883    |                          |       |  |  |
| 5  | 178.2508                       | 15.0873  | 3.700729 | 86.00116 | 123.0865 | 49.65524 | 0.885339 | 0.779329 | 0.564629 | 16.32779022    | 0.487866 | 0.847709 | 0.504878 | 0.061928    |                          |       |  |  |
| 6  | 152.4194                       | 69.78985 | 164.2814 | 85.36551 | 158.3874 | 89.97692 | 0.921873 | 0.403486 | 0.968231 | 34.14300495    | 0.433726 | 0.502993 | 0.475427 | 0.73995     |                          |       |  |  |
| 7  | 91.29353                       | 37.8373  | 88.94067 | 59.35629 | 184.3626 | 82.16292 | 0.06767  | 0.017572 | 0.181122 | 31.99869093    | 0.507584 | 0.894737 | 0.442552 | 0.331976    |                          |       |  |  |
| 8  | 3.700729                       | 86.00116 | 123.0865 | 49.65524 | 147.6414 | 64.49104 | 0.088999 | 0.504327 | 0.854576 | 40.9019159     | 0.71862  | 0.745158 | 0.547985 | 0.476308    |                          |       |  |  |
| 9  | 164.2814                       | 85.36551 | 158.3874 | 89.97692 | 35.25323 | 81.79743 | 0.92863  | 0.659347 | 0.076244 | 22.75953032    | 0.487097 | 0.151289 | 0.261195 | 0.683482    |                          |       |  |  |
| 10 | 88.94067                       | 59.35629 | 184.3626 | 82.16292 | 81.14124 | 66.02276 | 0.687263 | 0.322607 | 0.987039 | 8.444640274    | 0.807299 | 0.164574 | 0.369241 | 0.345177    |                          |       |  |  |
| 11 | 123.0865                       | 49.65524 | 147.6414 | 64.49104 | 187.0939 | 34.19706 | 0.635399 | 0.632667 | 0.809434 | 9.962979418    | 0.614349 | 0.06138  | 0.56351  | 0.425956    |                          |       |  |  |
| 12 | 158.3874                       | 89.97692 | 35.25323 | 81.79743 | 183.3809 | 28.97259 | 0.794765 | 0.830204 | 0.811405 | 35.22142855    | 0.029642 | 0.615448 | 0.834839 | 0.342569    |                          |       |  |  |
| 13 | 184.3626                       | 82.16292 | 81.14124 | 66.02276 | 82.05404 | 34.11936 | 0.153895 | 0.424834 | 0.883003 | 14.96950455    | 0.747397 | 0.280332 | 0.113633 | 0.759876    |                          |       |  |  |
| 14 | 147.6414                       | 64.49104 | 187.0939 | 34.19706 | 178.7299 | 53.4079  | 0.533081 | 0.301657 | 0.353248 | 30.29600206    | 0.865522 | 0.796403 | 0.155341 | 0.881378    |                          |       |  |  |
| 15 | 35.25323                       | 81.79743 | 183.3809 | 28.97259 | 11.57826 | 72.71132 | 0.41647  | 0.299607 | 0.620015 | 31.62317841    | 0.323331 | 0.165224 | 0.00238  | 0.56314     |                          |       |  |  |
| 16 | 81.14124                       | 66.02276 | 82.05404 | 34.11936 | 70.57363 | 30.92902 | 0.573268 | 0.062108 | 0.23141  | 46.62710597    | 0.189734 | 0.792796 | 0.354353 | 0.525509    |                          |       |  |  |
| 17 | 187.0939                       | 34.19706 | 178.7299 | 53.4079  | 162.6333 | 83.8496  | 0.652097 | 0.240897 | 0.053875 | 49.87296174    | 0.625249 | 0.006254 | 0.210735 | 0.386447    |                          |       |  |  |
| 18 | 183.3809                       | 28.97259 | 11.57826 | 72.71132 | 1.97226  | 56.80725 | 0.978891 | 0.143761 | 0.247022 | 34.69136141    | 0.431077 | 0.399617 | 0.828076 | 0.287765    |                          |       |  |  |
| 19 | 82.05404                       | 34.11936 | 70.57363 | 30.92902 | 27.77818 | 37.04136 | 0.4843   | 0.206074 | 0.49413  | 46.10168572    | 0.197564 | 0.743096 | 0.367147 | 0.766769    |                          |       |  |  |
| 20 | 178.7299                       | 53.4079  | 162.6333 | 83.8496  | 40.55304 | 70.27399 | 0.987911 | 0.231601 | 0.957599 | 27.37774707    | 0.786604 | 0.625877 | 0.885768 | 0.359594    |                          |       |  |  |
| 21 | 11.57826                       | 72.71132 | 1.97226  | 56.80725 | 39.74435 | 54.65712 | 0.490074 | 0.467727 | 0.572927 | 43.62310257    | 0.272455 | 0.393526 | 0.819832 | 0.228418    |                          |       |  |  |
| 22 | 70.57363                       | 30.92902 | 27.77818 | 37.04136 | 120.7585 | 44.48802 | 0.340818 | 0.577463 | 0.565399 | 38.43397192    | 0.145563 | 0.558775 | 0.915796 | 0.963056    |                          |       |  |  |
| 23 | 162.6333                       | 83.8496  | 40.55304 | 70.27399 | 70.27399 | 69.45672 | 0.019344 | 0.71138  | 0.221543 | 7.929441       | 0.253349 | 0.845703 | 0.827804 | 0.823957    |                          |       |  |  |
| 24 | 1.97226                        | 56.80725 | 39.74435 | 54.65712 | 39.76285 | 62.13101 | 0.646096 | 0.257552 | 0.681301 | 44.48211709    | 0.805569 | 0.263017 | 0.983299 | 0.573227    |                          |       |  |  |
| 25 | 27.77818                       | 37.04136 | 120.7585 | 44.48802 | 3.054785 | 79.48211 | 0.83359  | 0.785742 | 0.059935 | 39.40235173    | 0.110572 | 0.086256 | 0.980632 | 0.36251     |                          |       |  |  |
| 26 | 40.55304                       | 70.27399 | 54.43758 | 69.45672 | 149.3571 | 95.68434 | 0.50774  | 0.347406 | 0.312918 | 10.28973851    | 0.669194 | 0.740649 | 0.279433 | 0.853239    |                          |       |  |  |
| 27 | 39.74435                       | 54.65712 | 39.76285 | 62.13101 | 89.01929 | 52.25903 | 0.070657 | 0.402958 | 0.043994 | 30.33806979    | 0.397392 | 0.960242 | 0.283576 | 0.442731    |                          |       |  |  |
| 28 | 120.7585                       | 44.48802 | 3.054785 | 79.48211 | 186.3629 | 88.01422 | 0.907364 | 0.5379   | 0.83732  | 23.21173837    | 0.586714 | 0.375986 | 0.417972 | 0.753587    |                          |       |  |  |
| 29 | 54.43758                       | 69.45672 | 149.3571 | 95.68434 | 95.19887 | 17.29561 | 0.124853 | 0.252374 | 0.860886 | 16.7653966     | 0.625043 | 0.007553 | 0.653277 | 0.021514    |                          |       |  |  |
| 30 | 39.76285                       | 62.13101 | 89.01929 | 52.25903 | 83.72989 | 97.97469 | 0.84519  | 0.080212 | 0.62993  | 23.4929652     | 0.172592 | 0.053041 | 0.025871 | 0.561698    |                          |       |  |  |
| 31 | 3.054785                       | 79.48211 | 186.3629 | 88.01422 | 169.2443 | 27.14473 | 0.586656 | 0.335036 | 0.345026 | 19.20180997    | 0.786552 | 0.254478 | 0.559519 | 0.880253    |                          |       |  |  |
| 32 | 149.3571                       | 95.68434 | 95.19887 | 17.29561 | 105.0305 | 25.23293 | 0.148569 | 0.348968 | 0.674349 | 18.25184221    | 0.474312 | 0.751451 | 0.07117  | 0.58649     |                          |       |  |  |



In the next stage of experiment setting, we provide the image. It is this image which is to be protected using the GPKR watermarking technique. The image is a Lena picture. Such pictures have been most widely used as standard images to test the embedding and extraction process. The Lena is shown in Figure 6.17.



Figure 6.17 The Lena figure (<http://www.cs.cmu.edu/~chuck/lennapg/lenna.shtml>)

```

GPKR WATERMARKING TECHNIQUE
Copyright protection of Images in WMSNs
Applied Pyramid Transforms for reducing image
Generating Watermark Binary Matrix
Watermark Constraints and Kolmogorov Rule
Image reduction and Kolmogorov rule

Which one do you want = 1
Image that the multimedia Sensor node has captured? imread('len_std.jpg')
How many times will be it reduced? 6

n = 6

Matrix decimal from an image

ans =

 181 187 196 205 94 100 109 118 102 108 117 126
 179 185 195 203 92 98 108 116 100 106 116 124
 176 182 192 200 89 95 105 113 97 103 113 121
 172 179 188 196 85 92 101 109 93 100 109 117

Convert Matrix decimal to Matrix binary

st =

 1 0 1 1 0 1 0 1
 1 0 1 1 0 0 1 1
 1 0 1 1 0 0 0 0
 1 0 1 0 1 1 0 0
 1 0 1 1 1 0 1 1
 1 0 1 1 1 0 0 1
 1 0 1 1 0 1 1 0
 1 0 1 1 0 0 1 1
 1 1 0 0 0 0 1 0
 1 1 0 0 0 0 0 1
 1 1 0 0 0 0 0 0
 1 0 1 1 1 1 0 0

```

---

```

1 1 0 0 1 1 0 1
1 1 0 0 1 0 1 1
1 1 0 0 1 0 0 0
1 1 0 0 0 1 0 0
0 1 0 1 1 1 1 0
0 1 0 1 1 1 0 0
0 1 0 1 1 0 0 1
0 1 0 1 0 1 0 1
0 1 1 0 0 1 0 0
0 1 1 0 0 0 1 0
0 1 1 0 0 0 1 1
0 1 0 1 1 1 1 1
0 1 0 1 1 1 0 0
0 1 1 0 1 1 0 1
0 1 1 0 1 1 0 0
0 1 1 0 1 0 0 1
0 1 1 0 0 1 0 1
0 1 1 1 0 1 1 0
1 0 1 0 1 1 1 0
1 0 0 1 1 1 1 0
0 0 1 1 1 1 1 0
0 1 1 1 1 1 1 0
1 0 1 1 0 1 1 0
1 0 0 0 1 1 1 0
0 0 1 0 1 1 1 0
1 0 1 0 1 1 1 0
0 0 1 0 0 1 1 0
0 0 1 1 0 1 1 0
1 0 1 1 1 0 1 0
1 0 0 0 0 1 1 0
0 0 1 0 0 1 1 0
0 1 1 0 0 1 1 0
0 0 1 1 0 1 1 0
1 0 1 1 1 0 1 0
1 0 0 0 0 1 1 0
0 0 1 0 0 1 1 0
0 1 1 0 0 1 1 0

```

Watermark Matrix Signal to become 2 Rows x 1 column

Convert watermark constraint using Kolmogorov rule

-----

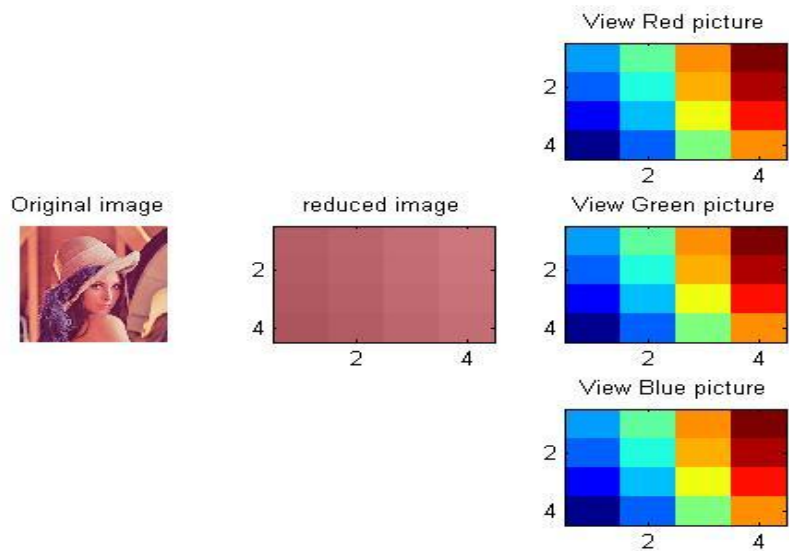


Figure 6.18 The process of reducing the Lena image

Figure 6.18 shows the process of reducing the Lena image.

We now generate watermark constraints as follows:

```

=====
 GPKR WATERMARKING TECHNIQUE
 Copyright protection of Images in WMSNs
 1. Dividing watermark using Kolmogorov rule
 2. Menu GPKR Watermarking technique
=====

Which one do you want = 1
Kolmogorov rule

1 2 3 4 5 6 7 8

x y z r s t u v

et eda edb edc d1 d2 d3 d4

Watermark constrains of the Lena image
How many rows do you want to create watermark constrints = 8
Watermark constraints are as follows
Watermark
Watermark =

d2 + d4 + edb + edc + et
d3 + d4 + edb + edc + et
edb + edc + et
d1 + d2 + edb + et
d1 + d3 + d4 + edb + edc + et
d1 + d4 + edb + edc + et
d2 + d3 + edb + edc + et
d3 + d4 + edb + edc + et
d2 + eda + et
d3 + d4 + eda + et
eda + et
d1 + d2 + edb + edc + et
d1 + d2 + d4 + eda + et
d1 + d3 + d4 + eda + et
d1 + eda + et
d2 + eda + et
d1 + d2 + d3 + eda + edc
d1 + d2 + eda + edc
d1 + d4 + eda + edc
d2 + d4 + eda + edc
d2 + eda + edb
d3 + eda + edb
d1 + d2 + d3 + d4 + eda + edc
d1 + d2 + eda + edc
d1 + d2 + d4 + eda + edb

```

---

```

d1 + d2 + eda + edb
d1 + d4 + eda + edb
d2 + d4 + eda + edb
d2 + d3 + eda + edb + edc
d2 + eda + edb + edc
d4 + eda + edb + edc
d1 + d2 + d4 + eda + edb
d1 + d2 + d3 + edb + et
d1 + d2 + d3 + edc + et
d1 + d2 + d3 + edb + edc
d1 + d2 + d3 + eda + edb + edc
d2 + d3 + edb + edc + et
d1 + d2 + d3 + et
d1 + d2 + d3 + edb
d1 + d2 + d3 + edb + et
d2 + d3 + edb
d2 + d3 + eda + edb + et
d2 + d3 + eda + edc
d2 + d3 + edb + edc
d1 + d3 + edb + edc + et
d2 + d3 + et
d2 + d3 + edb
d2 + d3 + eda + edb

```

The watermark constraints will be embedded into the cover medium. The next part explains how to expand the reduced image.

```

=====
 MAIN MENU FOR GENERATING WATERMARK
 GPKR WATERMARKING TECHNIQUE
 Extracting WMSNs
This Program applied Pyramid Transforms to expand THE image

Expanding should equal the reduced image
=====

st =

181 187 196 205 94 100 109 118 102 108 117 126
179 185 195 203 92 98 108 116 100 106 116 124
176 182 192 200 89 95 105 113 97 103 113 121
172 179 188 196 85 92 101 109 93 100 109 117

```

How many times do you want to expand the image? 6

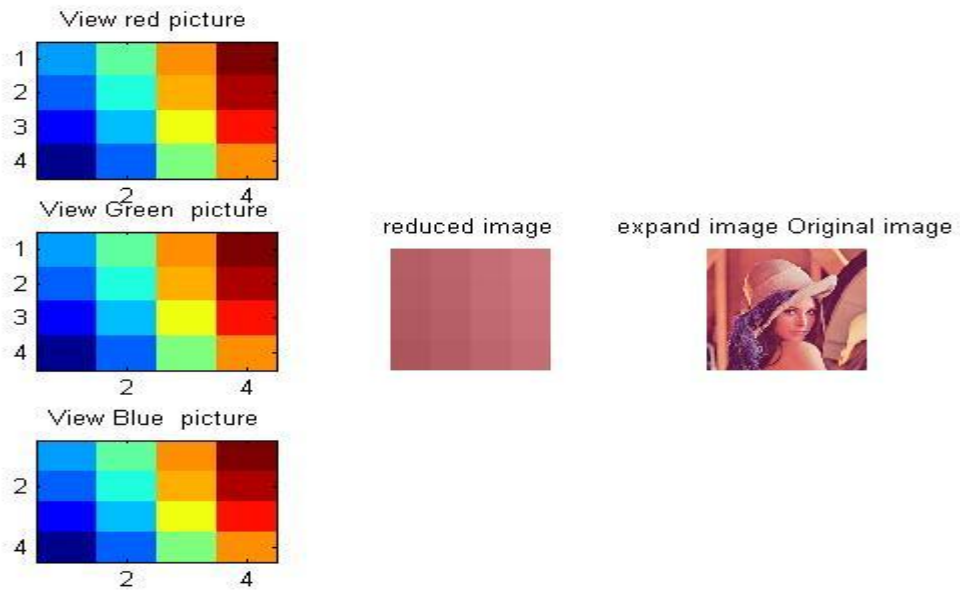


Figure 6.19 The Lena image expansion process

Figure 6.19 shows the Lena image expansion process

## 6.5.2 Parameters

Table 6.4 lists all the parameters and their associated values, used for capturing results from the algorithm.

Table 6.4 Parameters and their associated values used in the GPKR watermarking technique

| Parameter                                              | Description                                                                                                                           | Metric     |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|------------|
| N                                                      | Number of sensor node                                                                                                                 | Integer    |
| $(x_i, y_j)$<br>$i = j = 1, 2, \dots, n$               | Position of two-dimensional sensor networks                                                                                           | Coordinate |
| $T_c$                                                  | Temperature of the propagation media                                                                                                  | Degree     |
| $t_{DA}, t_{DB}, t_{DC}$                               | Time transmission between nodes D to A, D to B, and D to C                                                                            | Second     |
| $V_s$                                                  | Speed acoustic signal                                                                                                                 | (m/s)      |
| $\varepsilon_t$                                        | Error in the measurement of temperature                                                                                               | -          |
| $\varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}$ | Error in the measurement of the timer from D to A, D to B, and D to C                                                                 | -          |
| $\delta_1, \delta_2, \delta_3$                         | Error in measurement between the Euclidean measurement and that measured using time differences of optimal D to A, D to B, and D to C | -          |
| $\delta_4$                                             | Auxiliary variable                                                                                                                    | -          |
| $\tau_1, \tau_2, \tau_3, \tau_4$                       | The values are selected such that the feasibility of the solution space of the optimization problem is not harmed                     | -          |
| Image data                                             | Image sensed by a multimedia sensor node                                                                                              | Bit        |
| <i>threshold</i>                                       | Normalized correlation of the results of the errors of cover medium with watermark constraints                                        | -          |
| $sim(C', X')$                                          | Normalized correlation of the results of the error of cover medium with watermark constraint attacks                                  | -          |
| Watermark constraints                                  | Result from pseudo code                                                                                                               | -          |

### 6.5.3 Attack Characterization

This section discusses how WMSNs can be attacked in different ways and how the security scheme proposed by us thwarts these attacks. Consider the general model of a watermarking technique for copyright protection for WMSNs according to a communication formulation. Its block diagram is shown in Figure 6.1. We consider in detail the corresponding weaknesses for the GPKR watermarking technique that can be used by an attacker. Assuming that the results of the watermark constraints embedding process are estimated by the attacker and can be changed, modified and removed, the corresponding attacks are:

#### 6.5.3.1 Deletion of the Results of Embedding Watermark Constraints

In this type of attack the attacker drops a number of results of the watermark embedding process, hoping that the received results will not approximate to the results of the errors of watermark constraint embedding process and the receiver will not be able to expand the reduced image. Table 6.5 shows the results of data deletion attack.

Table 6.5 Results of data deletion attack

| No. | et       | eda      | edb      | edc      | d1       | d2       | d3       | d4 |
|-----|----------|----------|----------|----------|----------|----------|----------|----|
| 1   | -368.559 | 0.871006 | -0.69507 | -0.15576 | 1.1E-15  | 8.27E-15 | 0.022656 | 0  |
| 2   | -354.786 | 0.067108 | 0.016251 | -2.4E-15 | 0        | 0        | -2.4E-15 | 0  |
| 3   | -366.574 | 0.314889 | -0.20552 | -0.83653 | 1.46E-16 | 0        | 0        | 0  |
| 4   | -91715.1 | -4.14155 | -3.84053 | -4.25654 | 0.33915  | 0.01285  | 4.136199 | 0  |
| 5   | -1.9E+07 | -16.2954 | -16.1894 | -15.9747 | -1.2E-09 | 6.03E-09 | 0        | 0  |
| 6   | -5.1E+09 | -1.79293 | -1.27421 | -1.83895 | 0.051088 | 1.698487 | 0.000274 | 0  |
| 7   | -46.896  | 0.202836 | 0.03697  | 0.09217  | 0        | 0        | 0        | 0  |
| 8   | -9E+09   | -1.01671 | -1.43204 | -1.78229 | -2.9E-08 | 7.47E-09 | 8.27E-06 | 0  |
| 9   | -2.8E+09 | -2.2487  | -1.97941 | -1.39631 | 6.73E-10 | 0.000684 | 0.000813 | 0  |
| 10  | -2.8E+08 | -4.34846 | -3.9838  | -4.64823 | 1.56E-09 | 0        | 9.96E-12 | 0  |
| 11  | -415.921 | 0.352388 | -0.29101 | -0.29101 | -1.2E-17 | 0        | 0        | 0  |
| 12  | -1.3E+08 | -3.89156 | -3.927   | -3.9082  | 0.01307  | 3.54E-09 | -4.4E-11 | 0  |
| 13  | -132.978 | 0.436673 | -0       | -0.81604 | 0        | 0.166699 | 0        | 0  |
| 14  | -387.543 | 0.344457 | -0       | -0.11912 | -3.3E-14 | 0        | 1.61E-17 | 0  |
| 15  | -154.952 | 0.031489 | -0       | 0.6606   | 0        | 0.031489 | 0        | 0  |
| 16  | -3.4E+08 | -0.88494 | -0.3738  | -0.54307 | 0.22836  | 0.007178 | 0.492283 | 0  |
| 17  | -320.099 | -0.13532 | -0.18463 | 0.141574 | 0        | 0        | 0        | 0  |

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 18 | -7.2E+09 | -4.45188 | -3.61675 | -3.72001 | 0.008996 | 0.043237 | 0.048432 | 0 |
| 19 | -2.8E+09 | -2.92602 | -2.0478  | -2.93585 | 5.54E-08 | 0        | 0        | 0 |
| 20 | -498499  | -1.36528 | -0.68926 | -1.35542 | 0.000118 | 0        | 4.73E-12 | 0 |
| 21 | -413.476 | 0.344074 | -0.11566 | -0.11566 | -1.4E-15 | 2.66E-14 | 0        | 0 |
| 22 | -4.7E+09 | -4.30285 | -4.5395  | -4.52744 | 0        | 0        | 7.87E-12 | 0 |
| 23 | -2883228 | -9.5167  | -10.2117 | -9.72239 | 0.039146 | 0.971274 | 0.000217 | 0 |
| 24 | -6.6E+08 | -5.70813 | -5.31959 | -5.74334 | 0        | -1.7E-09 | 0        | 0 |
| 25 | -385.615 | -0.27625 | 5.93E-14 | 0.36251  | -1.4E-14 | 3.34E-14 | -7.5E-19 | 0 |
| 26 | -261.886 | 0.180011 | -1.8E-13 | 0.099421 | 0        | -4.4E-14 | 1.01E-17 | 0 |
| 27 | -89.9681 | 0.288105 | -0.23769 | -0.00453 | -7.4E-16 | -3.2E-15 | 8.76E-18 | 0 |
| 28 | -1.5E+09 | -7.08853 | -6.71907 | -7.01849 | -1.8E-07 | -1.4E-07 | 0        | 0 |
| 29 | 1.39E-17 | 0.007553 | -1.2E-16 | -0.71373 | -1.3E-17 | 0        | 0        | 0 |
| 30 | -357.668 | 0.106291 | -0.08042 | -0.27835 | 0        | 0.02717  | 1.45E-15 | 0 |
| 31 | -571.535 | -3.1E+08 | -3.3E+08 | -3.4E+08 | -4.9E-07 | 257.0846 | -2.4E-11 | 0 |
| 32 | -201.677 | 0.373617 | -0.30245 | -0.58946 | 8.33E-17 | 0.287014 | 5.62E-25 | 0 |

### 6.5.3.2 False Data Insertion for the Results of Embedding Watermark Constraints

In this type of attack, the attacker inserts some false yet convincing data into the results from the watermark embedding process, hoping to find new results of error of the watermark constraint embedding process that will map into the existing solution. Table 6.6 shows the results of false data insertion attack.

Table 6.6 Results of false data insertion attack

| No. | et       | eda      | edb      | edc      | d1       | d2       | d3       | d4 |
|-----|----------|----------|----------|----------|----------|----------|----------|----|
| 1   | -368.559 | 0.871006 | -0.69507 | -0.15576 | 1.1E-15  | 8.27E-15 | 0.022656 | 0  |
| 2   | -354.786 | 0.067108 | 0.016251 | -2.4E-15 | 0        | 0        | -2.4E-15 | 0  |
| 3   | -366.574 | 0.314889 | -0.20552 | -0.83653 | 1.46E-16 | 0        | 0        | 0  |
| 4   | -91715.1 | -4.14155 | -3.84053 | -4.25654 | 0.33915  | 0.01285  | 4.136199 | 0  |
| 5   | -1.9E+07 | -16.2954 | -16.1894 | -15.9747 | -1.2E-09 | 6.03E-09 | 0        | 0  |
| 6   | -5.1E+09 | -1.79293 | -1.27421 | -1.83895 | 0.051088 | 1.698487 | 0.000274 | 0  |
| 7   | -46.896  | 0.202836 | 0.03697  | 0.09217  | 0        | 0        | 0        | 0  |
| 8   | -9E+09   | -1.01671 | -1.43204 | -1.78229 | -2.9E-08 | 7.47E-09 | 8.27E-06 | 0  |
| 9   | -2.8E+09 | -2.2487  | -1.97941 | -1.39631 | 6.73E-10 | 0.000684 | 0.000813 | 0  |
| 10  | -2.8E+08 | -4.34846 | -3.9838  | -4.64823 | 1.56E-09 | 0        | 9.96E-12 | 0  |
| 11  | -415.921 | 0.352388 | -0.29101 | -0.29101 | -1.2E-17 | 0        | 0        | 0  |
| 12  | -1.3E+08 | 3.89156  | -3.927   | 3.9182   | 0.0307   | 3.54E-09 | 1.4E-11  | 0  |
| 13  | -132.178 | 0.436675 | -0.32304 | -0.81504 | 0        | 0.161699 | 0        | 0  |



|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 14 | -387.543 | 0.344457 | -0.18912 | -0.18912 | -3.3E-14 | 0        | 1.61E-17 | 0 |
| 15 | -154.952 | 0.031489 | -0.02911 | -0.0606  | 0        | 0.031489 | 0        | 0 |
| 16 | -3.4E+08 | -0.88494 | -0.3738  | -0.54307 | 0.22836  | 0.007178 | 0.492283 | 0 |
| 17 | -320.099 | -0.13532 | -0.18463 | 0.141574 | 0        | 0        | 0        | 0 |
| 18 | -7.1E+09 | -4.45281 | -3.61675 | -3.7200  | 0.08995  | 0.43237  | 0.04843  | 0 |
| 19 | -2.8E+09 | -2.92602 | -2.6478  | -2.93585 | 5.54E-08 | 0        | 0        | 0 |
| 20 | -498499  | -1.36528 | -0.60926 | -1.33542 | 0.000118 | 0        | 4.73E-12 | 0 |
| 21 | -413.476 | 0.344074 | -0.11566 | -0.11566 | -1.4E-15 | 2.66E-14 | 0        | 0 |
| 22 | -4.7E+09 | -4.30285 | -4.5395  | -4.52744 | 0        | 0        | 7.87E-12 | 0 |
| 23 | -2883228 | -9.5167  | -10.2117 | -9.72239 | 0.039146 | 0.971274 | 0.000217 | 0 |
| 24 | -6.6E+08 | -5.70813 | -5.31959 | -5.74334 | 0        | -1.7E-09 | 0        | 0 |
| 25 | -385.615 | -0.27625 | 5.93E-14 | 0.36251  | -1.4E-14 | 3.34E-14 | -7.5E-19 | 0 |
| 26 | -261.886 | 0.180011 | -1.8E-13 | 0.099421 | 0        | -4.4E-14 | 1.01E-17 | 0 |
| 27 | -89.9681 | 0.288105 | -0.23769 | -0.00453 | -7.4E-16 | -3.2E-15 | 8.76E-18 | 0 |
| 28 | -1.5E+09 | -7.08853 | -6.71907 | -7.01849 | -1.8E-07 | -1.4E-07 | 0        | 0 |
| 29 | 1.39E-17 | 0.007553 | -1.2E-16 | -0.71373 | -1.3E-17 | 0        | 0        | 0 |
| 30 | -357.668 | 0.106291 | -0.08042 | -0.27835 | 0        | 0.02717  | 1.45E-15 | 0 |
| 31 | -571.535 | -3.1E+08 | -3.3E+08 | -3.4E+08 | -4.9E-07 | 257.0846 | -2.4E-11 | 0 |
| 32 | -201.677 | 0.373617 | -0.30245 | -0.58946 | 8.33E-17 | 0.287014 | 5.62E-25 | 0 |

Results of some convincing number data insertion are shown in Table 6.7.

Table 6.7 New results of false data insertion

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 11 | -9.6E+09 | -1.8512  | -1.48655 | -2.15098 | 0        | 2.46E-07 | 0        | 0 |
| 12 | -1.3E+08 | -1.24105 | -1.23832 | -1.41508 | 0        | -7.5E-09 | 0        | 0 |
| 13 | -1.2E+08 | -1.3061  | -1.36174 | -1.3362  | 1539375  | 114142.6 | 587841.1 | 0 |
| 14 | -2.7E+09 | -1.1476  | -1.412   | -1.87687 | 0        | -1.6E-08 | 4.7E-07  | 0 |
| 15 | -1.6E+08 | 1.45655  | -1.21897 | -1.27556 | 562469.5 | 4799116  | 45258.39 | 0 |
| 16 | -1.8E+09 | -1.35223 | -1.287   | -1.60741 | 56765374 | 2082626  | 2082763  | 0 |
| 17 | -6.8E+08 | -1.3432  | -0.83204 | -1.00135 | -5.8E-09 | -9.5E-10 | -9.8E-10 | 0 |
| 18 | -4.6E+08 | -1.1628  | -0.7516  | -0.56458 | 4.07E-09 | -1.2E-08 | 1.76E-09 | 0 |
| 19 | -4739296 | -2.26086 | -1.51799 | -1.59013 | 110916.1 | 115844.1 | 115557.1 | 0 |
| 20 | -2.1E+09 | -2.63393 | -1.93231 | -2.2161  | 5.4E+08  | 962600.7 | 5763527  | 0 |

### 6.5.3.3 Modification of the Results of Embedding Watermark Constraints

In this type of attack, the attacker gains access to the data paths in the network and successfully interprets the traffic. Having read the data, the attackers alters it by modifying the results of the watermark constraint embedding process, hoping to find new results of the embedding process that will map into the existing solution. Results of data modification attack are shown in Table 6.8

Table 6.8 Results of data modification attack

| o. | et       | eda      | edb       | edc      | d1       | d2       | d3       | d4 |
|----|----------|----------|-----------|----------|----------|----------|----------|----|
| 1  | -368.559 | 0.871006 | -0.69507  | -0.15576 | 1.1E-15  | 8.27E-15 | 0.022656 | 0  |
| 2  | -354.786 | 0.067108 | 0.016251  | -2.4E-15 | 0        | 0        | -2.4E-15 | 0  |
| 3  | -366.574 | 0.314889 | -0.20552  | -0.83653 | 1.46E-16 | 0        | 0        | 0  |
| 4  | -91715.1 | -4.14155 | -3.84053  | -4.25654 | 0.33915  | 0.01285  | 4.136199 | 0  |
| 5  | -1.9E+07 | -16.2954 | -16.1894  | -15.9747 | -1.2E-09 | 6.03E-09 | 0        | 0  |
| 6  | -5.1E+09 | -1.79293 | -1.27421  | -1.83895 | 0.051088 | 1.698487 | 0.000274 | 0  |
| 7  | -46.896  | 0.202836 | 0.03697   | 0.09217  | 0        | 0        | 0        | 0  |
| 8  | -9E+09   | -1.01671 | -1.43204  | -1.78229 | -2.9E-08 | 7.47E-09 | 8.27E-06 | 0  |
| 9  | -2.8E+09 | -2.2487  | -1.97941  | -1.39631 | 6.73E-10 | 0.000684 | 0.000813 | 0  |
| 10 | -2.8E+08 | -4.34846 | -3.9838   | -4.64823 | 1.56E-09 | 0        | 9.96E-12 | 0  |
| 11 | -415.921 | 0.352388 | -0.29101  | -0.29101 | -1.2E-17 | 0        | 0        | 0  |
| 12 | -1.3E+08 | -3.89156 | -3.927    | -3.9082  | 0.01307  | 3.54E-09 | -4.4E-11 | 0  |
| 13 | -132.978 | 0.436673 | -0.32304  | -0.81604 | 0        | 0.166699 | 0        | 0  |
| 14 | -387.543 | 0.344157 | -0.18912  | -0.18912 | -3.3E-14 | 0        | 1.61E-17 | 0  |
| 15 | -154.952 | 0.031488 | -0.029101 | -0.0606  | 0        | 0.001499 | 0        | 0  |
| 16 | -3.4E+08 | -0.88494 | -0.3738   | -0.54307 | 0.22836  | 0.007178 | 0.492283 | 0  |
| 17 | -320.099 | -0.13532 | -0.18463  | 0.141574 | 0        | 0        | 0        | 0  |
| 18 | -7.2E+09 | -4.45188 | -3.61675  | -3.72001 | 0.008996 | 0.043237 | 0.048432 | 0  |
| 19 | -2.8E+09 | -2.92602 | -2.6478   | -2.93585 | 5.54E-08 | 0        | 0        | 0  |
| 20 | -498499  | -1.36528 | -0.60926  | -1.33542 | 0.000118 | 0        | 4.73E-12 | 0  |
| 21 | -413.476 | 0.344074 | -0.11566  | -0.11566 | -1.4E-15 | 2.66E-14 | 0        | 0  |
| 22 | -4.7E+09 | -4.30285 | -4.5395   | -4.52744 | 0        | 0        | 7.87E-12 | 0  |
| 23 | -2883228 | -9.5167  | -10.2117  | -9.72239 | 0.039146 | 0.971274 | 0.000217 | 0  |
| 24 | -6.6E+08 | -5.70813 | -5.31959  | -5.74334 | 0        | -1.7E-09 | 0        | 0  |
| 25 | -385.615 | -0.27625 | 5.93E-14  | 0.36251  | -1.4E-14 | 3.34E-14 | -7.5E-19 | 0  |
| 26 | -261.886 | 0.180011 | -1.8E-13  | 0.099421 | 0        | -4.4E-14 | 1.01E-17 | 0  |
| 27 | -89.9681 | 0.288105 | -0.23769  | -0.00453 | -7.4E-16 | -3.2E-15 | 8.76E-18 | 0  |
| 28 | -1.5E+09 | -7.08853 | -6.71907  | -7.01849 | -1.8E-07 | -1.4E-07 | 0        | 0  |
| 29 | 1.39E-17 | 0.007553 | -1.2E-16  | -0.71373 | -1.3E-17 | 0        | 0        | 0  |

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 30 | -357.668 | 0.106291 | -0.08042 | -0.27835 | 0        | 0.02717  | 1.45E-15 | 0 |
| 31 | -571.535 | -3.1E+08 | -3.3E+08 | -3.4E+08 | -4.9E-07 | 257.0846 | -2.4E-11 | 0 |
| 32 | -201.677 | 0.373617 | -0.30245 | -0.58946 | 8.33E-17 | 0.287014 | 5.62E-25 | 0 |

Results of some convincing number modifications are as shown in Table 6.9.

Table 6.9 Results of new data modification attack

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 11 | -1.3E+08 | -1.24105 | -1.23832 | -1.41508 | 0        | -7.5E-09 | 0        | 0 |
| 12 | -1.2E+08 | -1.3061  | -1.36171 | -1.3362  | 1539375  | 114142.6 | 587841.1 | 0 |
| 13 | -2.7E+09 | -1.14176 | -1.4127  | -1.87087 | 0        | -9.6E-08 | 4.77E-07 | 0 |
| 14 | -1.8E+08 | -1.45655 | -1.21897 | -1.27356 | 962489.5 | 4799116  | 45258.39 | 0 |
| 15 | -1.8E+09 | -1.3223  | -1.27    | -1.6071  | 557653.4 | 2082625  | 2082763  | 0 |
| 16 | -6.8E+08 | -1.5432  | -0.83204 | -1.00135 | -5.8E-09 | -9.5E-10 | -9.8E-10 | 0 |
| 17 | -4.6E+08 | -1.1628  | -0.7516  | -0.56458 | 4.07E-09 | -1.2E-08 | 1.76E-09 | 0 |
| 18 | -4739296 | -2.26086 | -1.51799 | -1.59013 | 110916.1 | 115844.1 | 115557.1 | 0 |
| 19 | -2.1E+09 | -2.63393 | -1.93231 | -2.2161  | 5.4E+08  | 962600.7 | 5763527  | 0 |
| 20 | -1.5E+09 | -2.18338 | -1.42707 | -2.15306 | 1.16E-12 | -1.2E-07 | -1.2E-07 | 0 |

### 6.5.3.4 Replication of the Results of Embedding Watermark Constraints

In this type of attack, data resulting from the watermark embedding process are replicated and the replicated data is inserted into the existing constraints. In this way, the attacker manipulates a particular part of the network, and may even disconnect it completely.

Table 6.10 Results of data replication attack

| No. | et       | eda      | edb      | edc      | d1       | d2       | d3       | d4 |
|-----|----------|----------|----------|----------|----------|----------|----------|----|
| 1   | -368.559 | 0.871006 | -0.69507 | -0.15576 | 1.1E-15  | 8.27E-15 | 0.022656 | 0  |
| 2   | -354.786 | 0.067108 | 0.016251 | -2.4E-15 | 0        | 0        | -2.4E-15 | 0  |
| 3   | -366.574 | 0.314889 | -0.20552 | -0.83653 | 1.46E-16 | 0        | 0        | 0  |
| 4   | -91715.1 | -4.14155 | -3.84053 | -4.25654 | 0.33915  | 0.01285  | 4.136199 | 0  |
| 5   | -1.9E+07 | -16.2954 | -16.1894 | -15.9747 | -1.2E-09 | 6.03E-09 | 0        | 0  |
| 6   | -5.1E+09 | -1.79293 | -1.27421 | -1.83895 | 0.051088 | 1.698487 | 0.000274 | 0  |
| 7   | -46.896  | 0.202836 | 0.03697  | 0.09217  | 0        | 0        | 0        | 0  |
| 8   | -9E+09   | -1.01671 | -1.43204 | -1.78229 | -2.9E-08 | 7.47E-09 | 8.27E-06 | 0  |
| 9   | -2.8E+09 | -2.2487  | -1.97941 | -1.39631 | 6.73E-10 | 0.000684 | 0.000813 | 0  |

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 10 | -2.8E+08 | -4.34846 | -3.9838  | -4.64823 | 1.56E-09 | 0        | 9.96E-12 | 0 |
| 11 | -415.921 | 0.352388 | -0.29101 | -0.29101 | -1.2E-17 | 0        | 0        | 0 |
| 12 | -1.3E+08 | -3.89156 | -3.927   | -3.9082  | 0.01307  | 3.54E-09 | -4.4E-11 | 0 |
| 13 | -132.978 | 0.436673 | -0.32304 | -0.81604 | 0        | 0.166699 | 0        | 0 |
| 14 | -387.543 | 0.344457 | -0.18912 | -0.18912 | -3.3E-14 | 0        | 1.61E-17 | 0 |
| 15 | -154.952 | 0.031489 | -0.02911 | -0.0606  | 0        | 0.031489 | 0        | 0 |
| 16 | -3.4E+08 | -0.38494 | -1.3738  | -0.51307 | -0.22836 | 0.007178 | 0.492283 | 0 |
| 17 | -320.99  | -0.13532 | -0.18463 | 0.14574  | 0        | 0        | 0        | 0 |
| 18 | -7.2E+09 | -4.45188 | -3.61675 | -3.72001 | 0.008996 | 0.043237 | 0.048432 | 0 |
| 19 | -2.8E+09 | -2.92602 | -2.6478  | -2.93585 | 5.54E-08 | 0        | 0        | 0 |
| 20 | -498499  | -1.36528 | -0.60926 | -1.33542 | 0.000118 | 0        | 4.73E-12 | 0 |
| 21 | -413.476 | 0.344074 | -0.11566 | -0.11566 | -1.4E-15 | 2.66E-14 | 0        | 0 |
| 22 | -4.7E+09 | -4.30285 | -4.5395  | -4.52744 | 0        | 0        | 7.87E-12 | 0 |
| 23 | -2883228 | -9.5167  | -10.2117 | -9.72239 | 0.039146 | 0.971274 | 0.000217 | 0 |
| 24 | -6.6E+08 | -5.70813 | -5.31959 | -5.74334 | 0        | -1.7E-09 | 0        | 0 |
| 25 | -385.615 | -0.27625 | 5.93E-14 | 0.36251  | -1.4E-14 | 3.34E-14 | -7.5E-19 | 0 |
| 26 | -261.886 | 0.180011 | -1.8E-13 | 0.099421 | 0        | -4.4E-14 | 1.01E-17 | 0 |
| 27 | -89.9681 | 0.288105 | -0.23769 | -0.00453 | -7.4E-16 | -3.2E-15 | 8.76E-18 | 0 |
| 28 | -1.5E+09 | -7.08853 | -6.71907 | -7.01849 | -1.8E-07 | -1.4E-07 | 0        | 0 |
| 29 | 1.39E-17 | 0.007553 | -1.2E-16 | -0.71373 | -1.3E-17 | 0        | 0        | 0 |
| 30 | -357.668 | 0.106291 | -0.08042 | -0.27835 | 0        | 0.02717  | 1.45E-15 | 0 |
| 31 | -571.535 | -3.1E+08 | -3.3E+08 | -3.4E+08 | -4.9E-07 | 257.0846 | -2.4E-11 | 0 |
| 32 | -201.677 | 0.373617 | -0.30245 | -0.58946 | 8.33E-17 | 0.287014 | 5.62E-25 | 0 |

Delete Replication

Results of some convincing new data replication are as shown in Table 6.9.

Table 6.11 Results of new data replication

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 11 | -9.6E+09 | -1.8512  | -1.48655 | -2.15098 | 0        | 2.46E-07 | 0        | 0 |
| 12 | -1.3E+08 | -1.24105 | -1.23832 | -1.41508 | 0        | -7.5E-09 | 0        | 0 |
| 13 | -1.2E+08 | -1.3061  | -1.36171 | -1.3362  | 1539375  | 114142.6 | 587841.1 | 0 |
| 14 | -2.7E+09 | -1.14176 | -1.4127  | -1.87087 | 0        | -9.6E-08 | 4.77E-07 | 0 |
| 15 | -1.8E+08 | -1.45655 | -1.21897 | -1.27356 | 962489.5 | 4799116  | 45258.39 | 0 |
| 16 | -9.6E+09 | -1.8512  | -1.48655 | -2.15098 | 0        | 2.46E-07 | 0        | 0 |
| 17 | -1.3E+08 | -1.24105 | -1.23832 | -1.41508 | 0        | -7.5E-09 | 0        | 0 |
| 18 | -1.2E+08 | -1.3061  | -1.36171 | -1.3362  | 1539375  | 114142.6 | 587841.1 | 0 |
| 19 | -2.7E+09 | -1.14176 | -1.4127  | -1.87087 | 0        | -9.6E-08 | 4.77E-07 | 0 |
| 20 | -1.8E+08 | -1.45655 | -1.21897 | -1.27356 | 962489.5 | 4799116  | 45258.39 | 0 |

Replication

## 6.6 Results and Observations

This section discusses the experiments carried out and the results observed on running our prototype. This is the sixth stage of the conceptual process described in Section 4.4.6, in which observations are made by analysing the results gathered from the prototype.

First we provide the results of the cover medium generation process. This process was undertaken by an embedder. In this process, we used the first row of Table 6.9 as the input to NLSP, as follows:

Table 6.12 The three positions, exact time and temperature

| No | Position of three sensor nodes |          |          |          |          |         |
|----|--------------------------------|----------|----------|----------|----------|---------|
|    | Xa                             | Ya       | Xb       | Yb       | Xc       | Yc      |
|    | 190.0259                       | 19.34312 | 97.19649 | 54.16739 | 91.29353 | 37.8373 |

| The exact time |          |          | temperature |
|----------------|----------|----------|-------------|
| Tda            | Tdb      | Tdc      | Tc          |
| 0.118738       | 0.845702 | 0.155561 | 8.354350415 |

The NLPS is as follows:

```
toms xd yd et eda edb edc d1 d2 d3 d4
f = et+ eda+edb+edc+d1+d2+d3+d4;
c ={
 -d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))* (tda+eda) <= d1
 -d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))* (tdb+edb) <= d2
 -d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))* (tdc+edc) <= d3
 0<=d1
 0<=d2
 0<=d3
 0<=d4 };
solution = ezsolve(f,c)
```

To solve equation 6.1, we run TOMLAB and get:

```
PKR WATERMARKING TECHNIQUE
1. Computation of Cover Medium
2. Checking results of this computation using Cover Medium
3. Menu GBKR Watermarking technique
=====
What Do you want = 1
How many experiments do you want = 1
Problem type appears to be: lpcon
Time for symbolic processing: 0.3743 seconds
Starting numeric solver
```



|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 14 | -1.8E+08 | -1.45655 | -1.21897 | -1.27356 | 962489.5 | 4799116  | 45258.39 | 0 |
| 15 | -1.8E+09 | -1.35223 | -1.287   | -1.60741 | 56765374 | 2082626  | 2082763  | 0 |
| 16 | -6.8E+08 | -1.3432  | -0.83204 | -1.00135 | -5.8E-09 | -9.5E-10 | -9.8E-10 | 0 |
| 17 | -4.6E+08 | -1.1628  | -0.7516  | -0.56458 | 4.07E-09 | -1.2E-08 | 1.76E-09 | 0 |
| 18 | -4739296 | -2.26086 | -1.51799 | -1.59013 | 110916.1 | 115844.1 | 115557.1 | 0 |
| 19 | -2.1E+09 | -2.63393 | -1.93231 | -2.2161  | 5.4E+08  | 962600.7 | 5763527  | 0 |
| 20 | -1.5E+09 | -2.18338 | -1.42707 | -2.15306 | 1.16E-12 | -1.2E-07 | -1.2E-07 | 0 |
| 21 | -1.4E+08 | -1.5288  | -1.50403 | -1.60774 | 1200849  | 126533.7 | 126534.3 | 0 |
| 22 | -5.3E+08 | -1.17576 | -1.41226 | -1.32107 | 101232.3 | 148138.5 | 24649266 | 0 |
| 23 | -3681655 | -1.39077 | -2.71227 | -1.88307 | 806427.6 | 781379.8 | 774145.4 | 0 |
| 24 | -4.1E+09 | -1.17725 | -0.78871 | -1.21246 | 7.14E-08 | -5.7E-12 | -5.2E-08 | 0 |
| 25 | -9.6E+09 | -1.50935 | -1.4615  | -0.73569 | 69794763 | 69794664 | 69794859 | 0 |
| 26 | -4.5E+08 | -0.67583 | -0.51549 | -0.48101 | 0        | 0        | 1.82E-12 | 0 |
| 27 | -2.2E+09 | -2.07901 | -1.56794 | -1.20901 | 1.12E+09 | 279434.4 | 233119.8 | 0 |
| 28 | -5E+09   | -1.88663 | -1.51717 | -1.81659 | 3.76E-09 | -3.7E-09 | 4.75E-07 | 0 |
| 29 | -3.6E+10 | -1.19362 | -1.32114 | -1.92966 | 0        | 0        | 0        | 0 |
| 30 | -8E+08   | -2.08051 | -1.09995 | -1.86525 | 85821095 | 24272349 | 85819407 | 0 |
| 31 | -3.7E+09 | -1.61236 | -1.36074 | -1.37073 | 0        | -8.4E-08 | 0        | 0 |
| 32 | -2.7E+07 | -1.03225 | -1.28812 | -1.5396  | 409971.6 | 435740.6 | 279742.7 | 0 |

In the next step, we provide the results of the watermark constraint embedding process. This process was also undertaken by an embedder. In this process, we used the first row of Table 6.3 as the input to NLSP, as follows:

Table 6.14 The three positions, exact time and temperature, and feasibility of the value

| No | Position of three sensor nodes |          |          |          |          |         | The exact time |          |          |
|----|--------------------------------|----------|----------|----------|----------|---------|----------------|----------|----------|
|    | Xa                             | Ya       | Xb       | Yb       | Xc       | Yc      | Tda            | Tdb      | Tdc      |
|    | 190.0259                       | 19.34312 | 97.19649 | 54.16739 | 91.29353 | 37.8373 | 0.118738       | 0.845702 | 0.155561 |

| temperature | The feasibility of value |          |          |          |
|-------------|--------------------------|----------|----------|----------|
| Tc          | thou1                    | thou2    | thou3    | thou4    |
| 8.354350415 | 0.181278                 | 0.796183 | 0.737905 | 0.715248 |

```

toms xd yd et eda edb edc d1 d2 d3 d4
f = et+ eda+edb+edc+d1+d2+d3+d4;
c ={
-d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))*(tda+eda) <= d1
-d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))*(tdb+edb) <= d2
-d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))*(tdc+edc) <= d3
d2 + d4 + edb + edc + et <=thou1
d3 + d4 + edb + edc + et <=thou2
edb + edc + et <=thou3

```

```

 d1 + d2 + edb + et <=thou4
d1 + d3 + d4 + edb + edc + et <=thou1
 d1 + d4 + edb + edc + et <=thou2
 d2 + d3 + edb + edc + et <=thou3
 d3 + d4 + edb + edc + et <=thou4
 d2 + eda + et <=thou1
 d3 + d4 + eda + et <=thou2
 eda + et <=thou3
 d1 + d2 + edb + edc + et <=thou4
 d1 + d2 + d4 + eda + et <=thou1
 d1 + d3 + d4 + eda + et <=thou2
 d1 + eda + et <=thou3
 d2 + eda + et <=thou4
 d1 + d2 + d3 + eda + edc <=thou2
 d1 + d2 + eda + edc <=thou3
 d1 + d4 + eda + edc <=thou4
 d2 + d4 + eda + edc <=thou2
 d2 + eda + edb <=thou1
 d3 + eda + edb <=thou2
d1 + d2 + d3 + d4 + eda + edc <=thou3
 d1 + d2 + eda + edc <=thou4
 d1 + d2 + d4 + eda + edb <=thou1
 d1 + d2 + eda + edb <=thou2
 d1 + d4 + eda + edb <=thou3
 d2 + d4 + eda + edb <=thou4
d2 + d3 + eda + edb + edc <=thou1
 d2 + eda + edb + edc <=thou2
 d4 + eda + edb + edc <=thou3
 d1 + d2 + d4 + eda + edb <=thou4
 d1 + d2 + d3 + edb + et <=thou1
 d1 + d2 + d3 + edc + et <=thou2
 d1 + d2 + d3 + edb + edc <=thou3
d1 + d2 + d3 + eda + edb + edc <=thou4
 d2 + d3 + edb + edc + et <=thou1
 d1 + d2 + d3 + et <=thou2
 d1 + d2 + d3 + edb <=thou3
 d1 + d2 + d3 + edb + et <=thou4
 d2 + d3 + edb <=thou1
d2 + d3 + eda + edb + et <=thou2
 d2 + d3 + eda + edc <=thou3
 d2 + d3 + edb + edc <=thou4
 d1 + d3 + edb + edc + et <=thou1
 d2 + d3 + et <=thou2
 d2 + d3 + edb <=thou3
 d2 + d3 + eda + edb <=thou4
0<=d1
0<=d2
0<=d3
0<=d4 };
solution = ezsolve(f,c)

```

To solve equation 6.1, we run TOMLAB and get:

```

=====
 GPKR WATERMARKING TECHNIQUE
 1. Embedding Watermark Constraints
 2. Main Menu GPKR Watermarking technique
=====
What Do you want = 1
How many experiment do you want = 1
Problem type appears to be: lpcon
Time for symbolic processing: 1.188 seconds
Starting numeric solver

```



```

=====** ===== * *
*
TOMLAB - Curtin University Ac. single user 501077. Valid to 2100-01-01
=====
==
Problem: --- 1: Problem 1 f_k -368.516004950792880000
 sum(|constr|) 0.000000000099486691
 f(x_k) + sum(|constr|) -368.516004950792880000
 f(x_0) 0.000000000000000000

Solver: snopt. EXIT=0. INFORM=1.
SNOPT 7.2-5 NLP code
Optimality conditions satisfied
FuncEv 1 ConstrEv 310 ConJacEv 310 Iter 95 MinorIter 102
CPU time: 0.156001 sec. Elapsed time: 0.156000 sec. solution =

 d1: 1.1033e-015
 d2: 8.2746e-015
 d3: 0.0227
 d4: 0
 eda: 0.8710
 edb: -0.6951
 edc: -0.1558
 et: -368.5588
 xd: 91.2935
 yd: 37.8373
The solution of this system is

```

| No | et       | eda      | edb      | edc      | d1      | d2       | d3       | d4 |
|----|----------|----------|----------|----------|---------|----------|----------|----|
|    | -368.559 | 0.871006 | -0.69507 | -0.15576 | 1.1E-15 | 8.27E-15 | 0.022656 | 0  |

We then compute all the input of Table 6.1 and get results as shown in Table 6.15.

Table 6.15 Results of watermark constraint embedding process

| No. | et       | eda      | edb      | edc      | d1       | d2       | d3       | d4 |
|-----|----------|----------|----------|----------|----------|----------|----------|----|
| 1   | -368.559 | 0.871006 | -0.69507 | -0.15576 | 1.1E-15  | 8.27E-15 | 0.022656 | 0  |
| 2   | -354.786 | 0.067108 | 0.016251 | -2.4E-15 | 0        | 0        | -2.4E-15 | 0  |
| 3   | -366.574 | 0.314889 | -0.20552 | -0.83653 | 1.46E-16 | 0        | 0        | 0  |
| 4   | -91715.1 | -4.14155 | -3.84053 | -4.25654 | 0.33915  | 0.01285  | 4.136199 | 0  |
| 5   | -1.9E+07 | -16.2954 | -16.1894 | -15.9747 | -1.2E-09 | 6.03E-09 | 0        | 0  |
| 6   | -5.1E+09 | -1.79293 | -1.27421 | -1.83895 | 0.051088 | 1.698487 | 0.000274 | 0  |
| 7   | -46.896  | 0.202836 | 0.03697  | 0.09217  | 0        | 0        | 0        | 0  |
| 8   | -9E+09   | -1.01671 | -1.43204 | -1.78229 | -2.9E-08 | 7.47E-09 | 8.27E-06 | 0  |
| 9   | -2.8E+09 | -2.2487  | -1.97941 | -1.39631 | 6.73E-10 | 0.000684 | 0.000813 | 0  |
| 10  | -2.8E+08 | -4.34846 | -3.9838  | -4.64823 | 1.56E-09 | 0        | 9.96E-12 | 0  |
| 11  | -415.921 | 0.352388 | -0.29101 | -0.29101 | -1.2E-17 | 0        | 0        | 0  |
| 12  | -1.3E+08 | -3.89156 | -3.927   | -3.9082  | 0.01307  | 3.54E-09 | -4.4E-11 | 0  |
| 13  | -132.978 | 0.436673 | -0.32304 | -0.81604 | 0        | 0.166699 | 0        | 0  |
| 14  | -387.543 | 0.344457 | -0.18912 | -0.18912 | -3.3E-14 | 0        | 1.61E-17 | 0  |
| 15  | -154.952 | 0.031489 | -0.02911 | -0.0606  | 0        | 0.031489 | 0        | 0  |

|    |          |          |          |          |          |          |          |   |
|----|----------|----------|----------|----------|----------|----------|----------|---|
| 16 | -3.4E+08 | -0.88494 | -0.3738  | -0.54307 | 0.22836  | 0.007178 | 0.492283 | 0 |
| 17 | -320.099 | -0.13532 | -0.18463 | 0.141574 | 0        | 0        | 0        | 0 |
| 18 | -7.2E+09 | -4.45188 | -3.61675 | -3.72001 | 0.008996 | 0.043237 | 0.048432 | 0 |
| 19 | -2.8E+09 | -2.92602 | -2.6478  | -2.93585 | 5.54E-08 | 0        | 0        | 0 |
| 20 | -498499  | -1.36528 | -0.60926 | -1.33542 | 0.000118 | 0        | 4.73E-12 | 0 |
| 21 | -413.476 | 0.344074 | -0.11566 | -0.11566 | -1.4E-15 | 2.66E-14 | 0        | 0 |
| 22 | -4.7E+09 | -4.30285 | -4.5395  | -4.52744 | 0        | 0        | 7.87E-12 | 0 |
| 23 | -2883228 | -9.5167  | -10.2117 | -9.72239 | 0.039146 | 0.971274 | 0.000217 | 0 |
| 24 | -6.6E+08 | -5.70813 | -5.31959 | -5.74334 | 0        | -1.7E-09 | 0        | 0 |
| 25 | -385.615 | -0.27625 | 5.93E-14 | 0.36251  | -1.4E-14 | 3.34E-14 | -7.5E-19 | 0 |
| 26 | -261.886 | 0.180011 | -1.8E-13 | 0.099421 | 0        | -4.4E-14 | 1.01E-17 | 0 |
| 27 | -89.9681 | 0.288105 | -0.23769 | -0.00453 | -7.4E-16 | -3.2E-15 | 8.76E-18 | 0 |
| 28 | -1.5E+09 | -7.08853 | -6.71907 | -7.01849 | -1.8E-07 | -1.4E-07 | 0        | 0 |
| 29 | 1.39E-17 | 0.007553 | -1.2E-16 | -0.71373 | -1.3E-17 | 0        | 0        | 0 |
| 30 | -357.668 | 0.106291 | -0.08042 | -0.27835 | 0        | 0.02717  | 1.45E-15 | 0 |
| 31 | -571.535 | -3.1E+08 | -3.3E+08 | -3.4E+08 | -4.9E-07 | 257.0846 | -2.4E-11 | 0 |
| 32 | -201.677 | 0.373617 | -0.30245 | -0.58946 | 8.33E-17 | 0.287014 | 5.62E-25 | 0 |

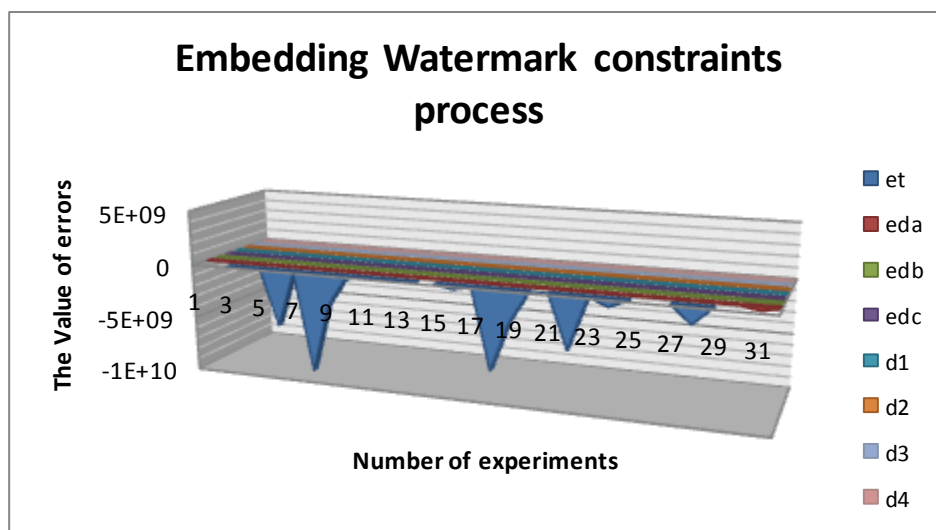


Figure 6.20 The value of error in watermark constraint embedding process

We next consider and evaluate in detail the corresponding attacks for GPKR watermarking technique that can be used by an attacker, as discussed in Section 6.5.3. There are four kinds of watermark constraint attacks. The results of detection of these attacks are given below:

### 6.6.1.1 Detection of the Result of Deletion Attack

As discussed in Section 6.5.3.1, the attacker can delete data by dropping a number of results of the embedding process. The process of detecting data deletion is as follows:

```

=====
GPKR WATERMARKING TECHNIQUE
1. Process of Detecting Deletion Attack
2. Menu GBKR Watermarking technique
=====
The value of threshold is = 8.578698153250081e+010

The value of similarity is = 9.626896789263084e+010
The watermark constraints change

```

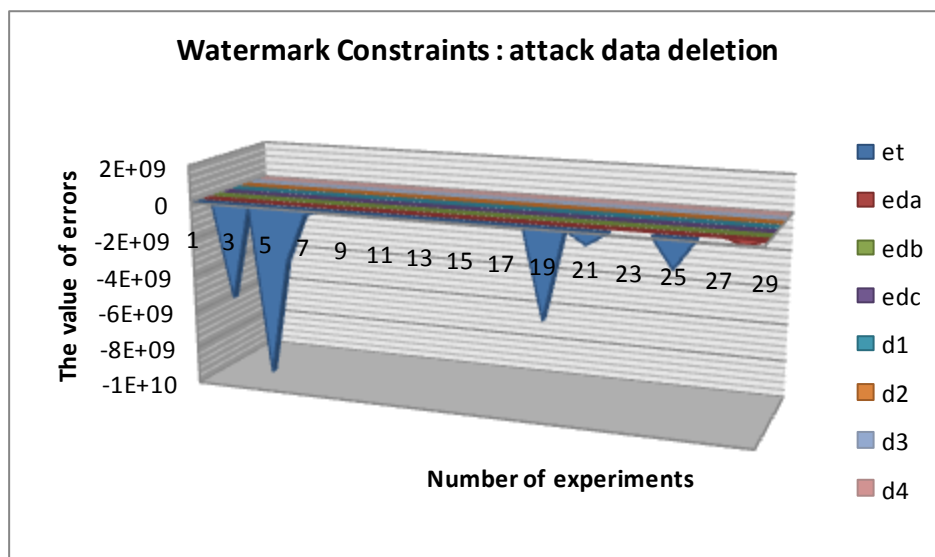


Figure 6.21 Value of the error of data deletion attack

### 6.6.1.2 Detection of the Result of False Data Insertion Attack

As discussed in Section 6.5.3.2, the attacker sets up false data by inserting a number of results of the embedding process. The process of detecting data insertion is as follows:

```

=====

```

GPKR WATERMARKING TECHNIQUE

1. Process of Detecting False Data Insertion
2. Menu GBKR Watermarking technique

```

=====
The value of threshold is = 8.578698153250081e+010

The value of similarity is 7.850046307403270e+010
False data insertion does not change the watermark

```

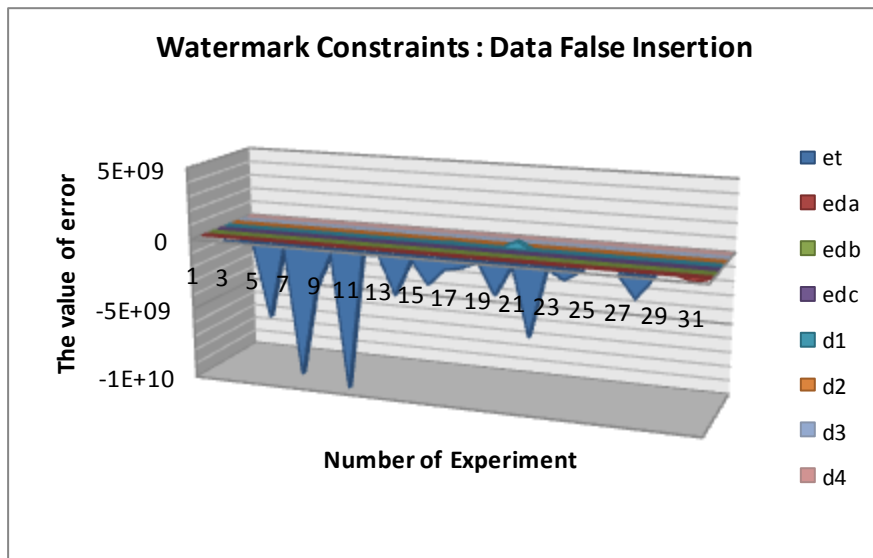


Figure 6.22 Value of the error of false data insertion attack

### 6.6.1.3 Detection of the Result of Modification Attack

As discussed in Section 6.5.3.4, the attacker alters data by modifying the results of the embedding process. The process of detecting data modification is as follows:

```

=====
GPKR WATERMARKING TECHNIQUE
1. Process of Detecting Data Modification
2. Menu GBKR Watermarking technique
=====
The value of threshold is = 8.578698153250081e+010

```

The value of similarity is 8.659658844907098e+010

The watermark constraints change

-----

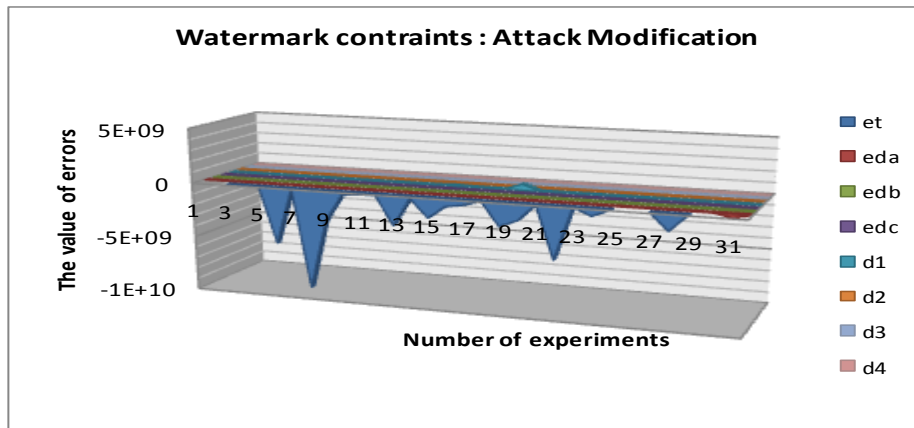


Figure 6.23 Value of the error of data modification attack

### 6.6.1.4 Detection of the Result of Replication Attack

As discussed in Section 6.5.3.4, the attacker replicates a number of results of the embedding process. The process of detecting data replication is as follows:

=====

GPKR WATERMARKING TECHNIQUE

1. Process of Detecting Data Replication
2. Menu GBKR Watermarking technique

=====

The value of threshold is = 8.578698153250081e+010

-----

The value of similarity is 3.108274290703391e+010

Replication attack does not change the watermark

-----

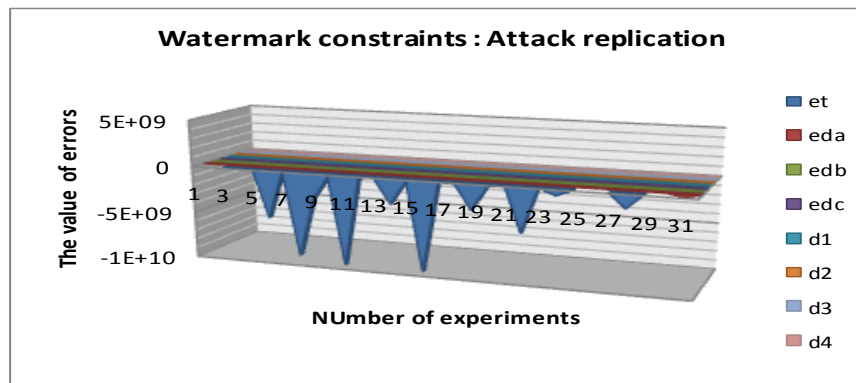


Figure 6.24 Value of the error of replication attack

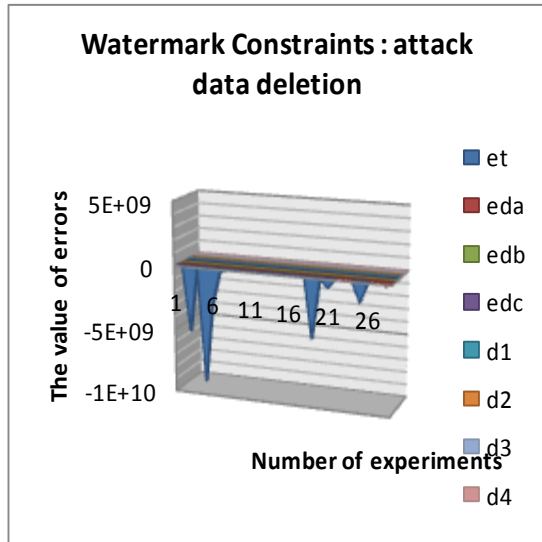
## 6.7 Validation and Discussion

Herein, a robust watermarking technique, termed GPKR watermarking technique, was proposed. We embedded the watermark constraints into the cover medium NLSP. The embedding process was based on equation constraints. The image captured by the WMSN nodes was reduced using pyramid transforms to get a reduced image. The reduced image was then quantified to get a decimal matrix. The decimal matrix was then converted into binary matrix. Kolmogorov rule was used to generate the watermark constraints by matching the bit numbers with the corresponding variable numbers. The variables in the group assigned the bit one were included in the linear while the variables in the group assigned the bit zero were not included. By testing several watermark constraint attacks, we found that the proposed technique worked well in the cover medium NLSP. For verifying whether the watermark constraints were present, the normalized difference error from the optimal solution between the watermarked solution and the solution obtained without watermark was used. The threshold measure was from the normalized correlation coefficient between the normalized difference error from the optimal solution between the watermarked solution  $X'$  and the solution obtained without watermark. The threshold was measured by

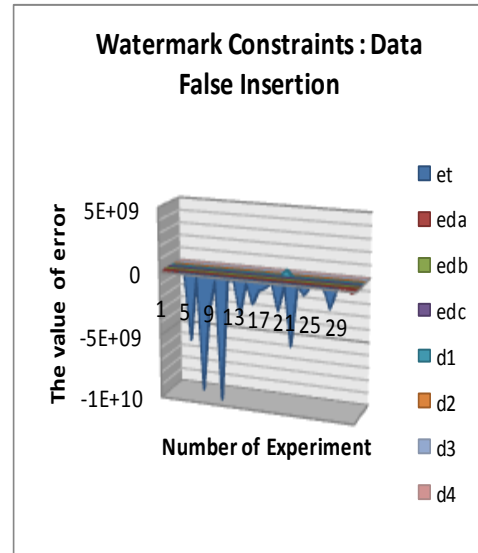
$$\frac{C * X'}{\sqrt{X' * X'}} \text{ where } C = X' - X. \text{ The similarity measure was obtained from the normalized}$$

correlation coefficient between the normalized difference error from the optimal solution between the watermarked solution  $X'$  and the solution obtained with watermark constraint attacks  $X''$ . The

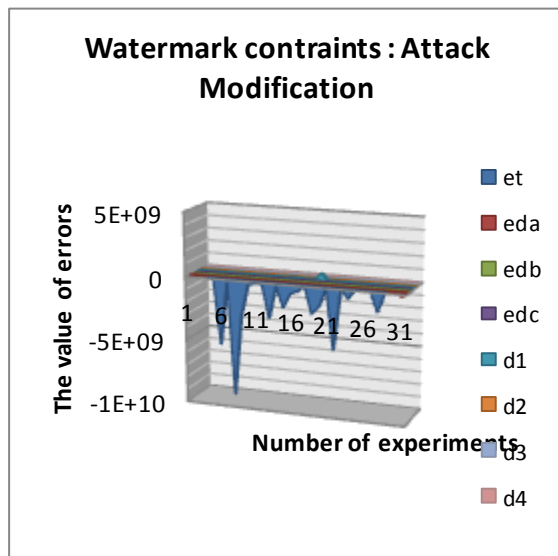
similarity was measured by  $\frac{C' * X'}{\sqrt{X'' * X''}}$ , where  $C' = X'' - X'$ . The results of all watermark constraint attacks are shown in Figure 6.25.



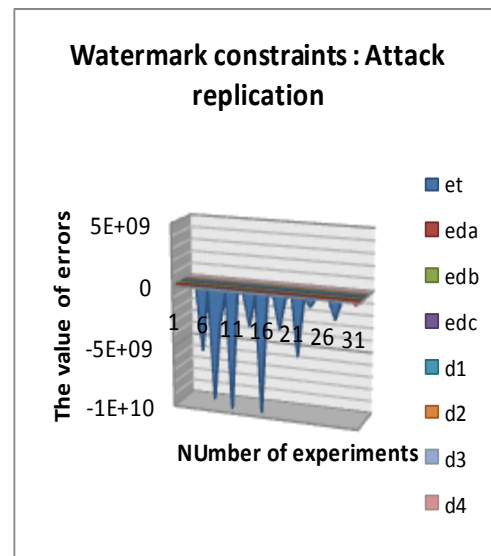
a. Data deletion attack



b. False data insertion attack



c. Data modification attack



d. Data replication attack

Figure 6.25 All watermark constraint attacks

---

From the experiments carried out, we make following observations:

1. In data deletion attack, data deletion changes the watermark constraints (as shown in Figure 6.26.a), because  $8.578698153250081e+010 < 9.626896789263084e+010$ . This means that the watermark constraints are not robust against data deletion, so the reduced image cannot be extracted or expanded.
2. In false data insertion attack, false data insertion changes the watermark constraints (as shown in Figure 6.26.b), because  $8.578698153250081e+010 < 7.850046307403270e+010$ . It means that the watermark constraints are robust against false data insertion, so the reduced image can be extracted and expanded.
3. In data modification attack, data modification changes the watermark constraints (as shown in Figure 6.26.c), because  $8.578698153250081e+010 < 8.659658844907098e+010$ . This means that the watermark constraints are not robust against false data insertion, so the reduced image cannot be extracted or expanded.
4. In data replication attack, data replication does not change the watermark constraints (as shown in Figure 6.26.d), because  $8.578698153250081e+010 > 3.108274290703391e+010$ . This means that the watermark constraints are robust against data replication, so the reduced image can be extracted and expanded.

Thus, our model is robust against insertion of a number of results of different watermark constraints, and replication of different watermark constraints. However, it is not robust against other watermark constraint attacks, such as deletion of a number of results of watermark constraints, or modification of a number of results of watermark constraints.

## 6.8 Comparative Analysis

This is the last stage of the conceptual process described in section 4.5.6, in which the proposed solution is compared with the existing solutions from the literature which we discussed in Chapter 2. Here, we carry out a comparative analysis of our technique with other techniques proposed by different researchers, as shown in Table 6.16.

---



Table 6.16 A comparative analysis with other approaches to copyright protection in WMSNs

| Kind of attacks      | Honggang, et.al<br><br>Honggang Wang, Dongming Peng , and Wei Wang 2008) | Pingping et.al<br><br>(Pingping, Yao Jiangtao, and Zhang Ye 2009) | Wang et.al<br><br>(Wang 2010) | <b>Padmavathi</b><br><br>(Padmavathi, Shanmugapriya, and Kalaivani 2010), | Kaur, S<br><br>(Kaur 2010 ) | <b>Masood, et al</b><br><br>(Masood, Haider, and Sadiq ur 2010) | <b>Harjito, B (2012)</b> |
|----------------------|--------------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------|--------------------------|
| Data deletion        | x                                                                        | x                                                                 | x                             | x                                                                         | x                           | x                                                               | X                        |
| False data insertion | x                                                                        | x                                                                 | x                             | x                                                                         | x                           | x                                                               | √                        |
| Data Modification.   | x                                                                        | x                                                                 | x                             | x                                                                         | x                           | x                                                               | x                        |
| Packet replication   | x                                                                        | x                                                                 | x                             | x                                                                         | x                           | x                                                               | √                        |

We compare 7 approaches in terms of false data insertion, data deletion, replication, and data modification. Honggang, et al. (Honggang Wang, Dongming Peng, and Wei Wang 2008), Wang (2010), Pingping, Yao Jiangtao, and Zhang Ye (2009), Padmavathi et al. (Padmavathi, Shanmugapriya, and Kalaivani 2010), Kaur (2010 ) and Masood et al. Masood, Haider, and Sadiq-ur (2010) do not provide protection against data deletion, false data insertion, data replication, and data modification. Our approach provides copyright protection to images against false data insertion and data replication. However, it is not robust against data deletion and data modification.

## 6.9 Conclusion

This chapter introduced and formulated a GPKR watermarking technique for copyright protection of multimedia data, or images, in WMSNs. The fundamental advantage of our approach is that it shares the results of the watermark signal in detection, as the watermark constraint attacks can be generated by modifying the results of watermark signal. However, our approach is robust against only 2 types of attacks. Therefore, we still need to improve our technique to make it more robust, considering various circumstances in which attackers launch different kinds of attacks.

# CHAPTER SEVEN

## CONCLUSION AND FUTURE WORK

This chapter covers:

- ▶ the current issues and problems with watermarking technique in WSNs and WMSNs,
- ▶ contribution of this research,
- ▶ conclusion of the thesis and indications for future work in the field.

### 7.1 Introduction

Watermarking techniques have been shown to be useful in addressing issues, such as copyright protection, tamper detection, content authentication and integrity. A number of researchers have studied watermarking techniques in the field of multimedia. However, in spite of their widespread use in the field of multimedia, it is often not feasible to adopt them directly to WSNs or WMSNs.

Several proposals associated with watermarking technique in WSNs and WMSNs have made significant advances in addressing the issue of secure communication, authentication, integrity and copyright protection. However, it is evident from a survey of the literature in the field that very few of them have addressed all the problems in their entirety.

The main purpose of this research was to address and develop a watermarking technique for WSNs and WMSNs, as the security mechanisms based on traditional cryptography algorithms are very expansive in terms of storage and energy consumption, as well as unsafe in terms of secure invisible communication and copyright protection.

This chapter provides a summary of the issues and problems addressed by the study, along with its main contributions. The next section elaborates the main problems with which the study was concerned.

---

## 7.2 Problems and Issues

This thesis addressed two main challenges associated with watermarking techniques in WSNs and WMSNs. These are:

- Problems with copyright protection of scalar data in WSNs
- Problems with copyright protection of images in WMSNs

In the following section, we summarize the problems and issues we faced while developing the solution and then highlight the main contribution of this thesis. We first discuss the main problems we addressed, followed by how we solve them.

### 7.2.1 Problems with Copyright Protection of Scalar Data in WSNs

As mentioned in Chapter 1 and 2, copyright protection of scalar data in WSNs has become a major security concern as both governments and businesses in Indonesia, as well as the world, are losing billions of Indonesian rupiah (IDR) every year due to scalar data theft or unauthorized use. Watermarking technique can help by adding copyright protection to identify the real owner of the content. However, currently even the most advanced watermarking techniques prove to be very fragile when used in WSNs. The main socio-economic and technical problems with the existing solutions, outlined in Chapter 3, include:

- Inability to securely add or embed a robust watermark signal to the data sent by a WSN deployed in hostile, unattended environments.
- Inability to make the WSN sensed data robust against severe watermark constraint attacks, such as modification, manipulation, Sybil and forwarding attack, during data transmission through a communication channel.

Thus, there was an urgent need for a more reliable and robust watermarking technique. Hence, we decided to develop an advanced watermarking technique for WSNs to address these technical difficulties.

---

---

## 7.2.2 Problems with Copyright Protection of Images in WMSNs

As mentioned in Chapter 1, copyright protection of images captured by WMSNs has become a major security concern for the commercial entities dealing with multimedia data. They need to ensure that the footages they own can be identified and the ownership can be proved in a court of law. Watermarking technique can help in it by adding a watermark to the images which is difficult to remove and can be used to identify the real owner of the content. However, as mentioned in Chapter 2, currently even the most advanced watermarking techniques prove to be very fragile in WMSNs. The main socio-economic and technical problems associated with the existing solutions, outlined in Chapter 3, are as follows:

- Inability to explain when and where the processes of embedding and extracting are to be undertaken.
- Inability to add or embed a robust watermark to multimedia data captured by a WMSN deployed in hostile, unattended environments.
- Inability to make the multimedia WMSN data robust against severe watermark constraint attacks, such as modification, manipulation, Sybil and forwarding attack, during data transmission through a communication channel.

Thus, there was a need for a more reliable and robust watermarking technique. Hence, we decided to develop an advanced watermarking technique for WMSNs to address these technical difficulties.

## 7.3 Contributions of the Thesis

This thesis contributed to two major areas of watermarking technique in WSNs and WMSNs which are increasingly becoming important, both socially and economically. It carried out:

- a comprehensive survey of watermarking techniques for WSNs, and
- a comprehensive survey of watermarking techniques for WMSNs.

To address the problems outlined in Chapter 3, the thesis proposed and evaluated the following solutions:

---

- 
- LKR watermarking technique for copyright protection of scalar data in WSNs.
  - GPKR watermarking technique for copyright protection of images in WMSNs.

### **7.3.1.1 Contribution 1: A State-of-the-art Watermarking Technology for WSNs**

The thesis first gave an overview of the literature and investigated the efforts made so far in the field of watermarking techniques for WSNs, as well as evaluated them on different parameters of digital watermarking techniques for WSNs.

The study evaluated 11 different works and made a comparative study on 10 different parameters representing the key aspects of watermarking technique.

Technically, the key parameters are: (1) cover medium, (2) sensed data (3) watermark generation, (4) types of watermark, (5) watermark key, (6) watermark embedding technique, (7) watermark detecting technique, (8) attack, (9) noise, and (10) transform domain. Based on these parameters, the evaluation is summarized below.

Most of the approaches in literature use ‘packet data’ as the cover medium and ‘text message’, as the watermark message which is converted into binary stream. Most of them use ‘sensed data’ as the message.

Binary stream has also been used as the watermark key by most of these approaches and the hash function for watermark generation, e.g. MD5 or SHA. Further, in most of the works, the ‘noise’ has not been mentioned. The ‘man-in-middle attack’ has been used by most as the vulnerability attack, e.g. false data insertion, data modification, forgery, or impersonation. ‘Least Significant Bits’ (LSB) has been used as the embedding technique by most, and finally, ‘statistic correlation’ has been used as the detecting technique, e.g. similarity, probability, mean deviation, standard deviation, and Gaussian hypothesis

---

---

### 7.3.1.2 Contribution 2: A State-of-the-art Watermarking Technology for WMSNs

The second important contribution of this thesis is investigating and providing a detailed insight into currently the most advanced watermarking techniques for WMSNs, as well as evaluating them on different parameters of watermarking techniques for WMSNs. We evaluated 6 existing works on watermarking technique for WMSNs on 10 different parameters, viz. (1) cover medium, (2) sensed data (3) watermark generator, (4) types of watermark, (5) watermark key, (6) watermark embedding technique, (7) watermark detecting technique, (8) attack, (9) noise, and (10) transform domain. Based on these parameters, the evaluation is summarized below:

Most of these approaches use ‘packet data’ as the cover medium, and ‘image’ as the sensed data. Most of them use ‘signal’ as the watermark, and the ‘two adaptive threshold’ as the watermark key.

Most of the works do not mention the watermark generator used. Two of these use the ‘two filter adaptive threshold’ as the inserting technique. Further, most of them use the ‘statistic approach’ as detecting technique, e.g. normalized correlation, and ‘dropped packet data’ as the noise. Finally, most of these approaches use ‘accidental’ vulnerable attacks, such as cropping and compressing

### 7.3.1.3 Contribution 3: LKR watermarking technique

The thesis developed the LKR watermarking technique for copyright protection of scalar sensed data, presented in Chapter 5. It developed a robust watermarking scheme that operates in the spatial domain. The main features of this scheme are as follows:

- a. It can operate on a minimum of 75 nodes over a 500 square meter area.
  - b. It uses a cover medium generated by using the theory of atomic trilateration.
  - c. It embeds message sensed data generated by LFSR.
  - d. It embeds watermark constraints generated by LFSR and Kolmogorov rule.
  - e. It is robust against packet replication and Sybil attacks, although not against false data insertion and data deletion.
-

This solution can be useful in reducing electricity theft in residential areas by automatically reporting it to the State Electricity Company .

### **7.3.1.4 Contribution 4: GPKR watermarking technique**

The thesis developed the GPKR watermarking technique for copyright protection of images in WMSNs, presented in Chapter 6. It developed a robust watermarking scheme that operates in spatial domain. The main features of this scheme are follows:

- a. It can operate on a minimum of 50 nodes randomly placed within 200 meter length and 100 meter width.
- b. It uses the Gaussian pyramid transforms to reduce the sensory image in order to produce the reduced image in an RGB colour.
- c. It embeds the watermark constraints generated using Kolmogorov rule.
- d. It provides copyright protection to images against false data insertion and packet replication, although not against data deletion and data modification.

This technique can be used to prove the ownership of images by inserting an invisible or visible watermark when a WMSN node captures the image.

## **7.4 Future Works**

The work undertaken in this thesis has been presented in peer reviewed international conferences. Over the course of the research, 5 papers were published in different international conferences, as mentioned in the Appendix. However, although a significant amount of effort has gone into this research, there is still scope for further improvements. The future research may aim at the following:

- LKR watermarking technique: Since this technique is robust against only two types of attacks, there is a need for further research to improve it, so as to be robust considering various other circumstances in which attackers carry out attacks.
-

- GPKR watermarking technique: Since this technique is robust against only two types of attack, i.e. false data insertion and packet replication, there is a need for further research to improve it, so as to be robust considering various other circumstances in which attackers carry out attacks.
  - Development of fragile watermarking technique for tampering detection in scalar WSN data and multimedia WMSN data. Embedding a fragile watermark can be used to detect the digital content in WSNs and WMSNs.
  - Development of a watermarking technique for data authentication in WSNs and WMSNs. This technique would verify whether the data has been exchanged with some malicious entry data that can be used to insert undesirable or unauthorized content, or send the same data back into the network.
-



---

## BIBLIOGRAPHY

- A. Herner, S. March, and S. Ram J. Park. 2004. "Design Science in Information System Research." *MIS Quarterly* no. Vol 28 (1): 75-105, 2004.
- Abdul Hadi Fikri Bin Abdul Hamid, Rozeha A. Rashid, Norsheila Fisal, S. K. S. Yusof, S. H. S. Ariffin Liza Latiff. 2009. "Development of IEEE802.15.4 based Wireless Sensor Network Platform for Image Transmission." *International Journal of Engineering & Technology IJET* no. 9 (10).
- Adelson, Edward H, Charles H Anderson, Bergen, James R, and Peter J Burt, Ogden, Joan M. 1984. "Pyramid methods in image processing." *RCA engineer* no. 29 (6):33-41.
- Adrian, Perrig Robert, Szewczyk J. D. Tygar Victor, and E. Culler Wen David. 2002. "SPINS: security protocols for sensor networks." *Wirel. Netw.* no. 8 (5):521-534. doi: <http://dx.doi.org/10.1023/A:1016598314198>.
- Agah, Afrand, Kalyan Basu, and Sajal K. Das. 2006. "Security enforcement in wireless sensor networks: A framework based on non-cooperative games." *Pervasive and Mobile Computing* no. 2 (2):137-158. doi: DOI: 10.1016/j.pmcj.2005.12.001.
- Ahmad, A., and A. M. Samad. 2010. Aerial mapping using high resolution digital camera and unmanned aerial vehicle for Geographical Information System. Paper read at Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on, 21-23 May 2010.
- Ahmed, Mohiuddin, Srikanth Krishnamurthy, Randy Katz, and Son Dao. 2002. "Trajectory control of mobile gateways for range extension in ad hoc networks." *Computer Networks* no. 39 (6):809-825. doi: Doi: 10.1016/s1389-1286(02)00249-9.
- Ajay Jangra, Swati. 2010 "Wireless Sensor Network (WSN): Architectural Design issues and Challenges." (*IJCSE*) *International Journal on Computer Science and Engineering* no. 02 (09):3089-3094.
- Akan, Ozgur B. Frossard, Pascal Zhang, and Nikil Qian Jayant. 2008. "Special issue on wireless multimedia sensor networks." *Computer Networks* no. 52 (13):2529-2531.
- Akyildiz, I. F., T. Melodia, and K. R. Chowdhury. 2008. "Wireless Multimedia Sensor Networks: Applications and Testbeds." *Proceedings of the IEEE* no. 96 (10):1588-1605.
- Akyildiz, Ian F., Tommaso Melodia, and Kaushik R. Chowdhury. 2007. "A survey on wireless multimedia sensor networks." *Computer Networks* no. 51 (4):921-960. doi: DOI: 10.1016/j.comnet.2006.10.002.
-

- 
- Akyildiz, Ian F. Melodia, Tommaso Chowdhury, Kaushik R. 2007. "A survey on wireless multimedia sensor networks." *Computer Networks* no. 51 (4):921-960.
- Akyildiz, Ian F., Dario Pompili, and Tommaso Melodia. 2005. "Underwater acoustic sensor networks: research challenges." *Ad Hoc Networks* no. 3 (3):257-279. doi: DOI: 10.1016/j.adhoc.2005.01.004.
- Al Ameen, Moshaddique, Jingwei Liu, and Kyungsup Kwak. 2012. "Security and privacy issues in wireless sensor networks for healthcare applications." *Journal of medical systems* no. 36 (1):93-101.
- Almalkawi, Islam T., Manel Guerrero Zapata, and Jamal N. Morillo-Pozo Al-Karaki, Julian. 2010. "Wireless Multimedia Sensor Networks: Current Trends and Future Directions." *Sensors* no. 10 6662 - 6717. doi: 10.3390/s100706662.
- Bai, Baochun, Janelle Harms, and Yuxi Li. 2008. "Configurable active multicast congestion control." *Computer Networks* no. 52 (7):1410-1432. doi: DOI: 10.1016/j.comnet.2007.12.010.
- Bartosz, Przydatek, Song Dawn, and Perrig Adrian. 2003. SIA: secure information aggregation in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*. Los Angeles, California, USA: ACM.
- BBC, News, and South Asia. 2012 "How drones work " <http://www.bbc.co.uk/news/world-south-asia-10713898> no. 31 January 2012.
- Castelluccia, C., E. Mykletun, and G. Tsudik. 2005. Efficient aggregation of encrypted data in wireless sensor networks. Paper read at Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on.
- Chen, Xiangqian, Kia Makki, and Kang Yen, Pissinou, Niki. 2009. "Sensor network security: a survey." *Communications Surveys & Tutorials, IEEE* no. 11 (2):52-73.
- Choudhary, Binod Kumar, navin Kumar Sinha, and Prem Shanker. 2012. "PYRAMID METHOD IN IMAGE PROCESSING."
- Cox, I. J., J. Kilian, and T. Shanon Leighton, T. 1997. "Secure spread spectrum watermarking for multimedia." *IEEE transactions on image processing* no. 6 (12):1673.
- de los Angeles Cosio Leon, M. Hipolito, J. I. N. Garcia, J. L. 2009. A Security and Privacy Survey for WSN in e-Health Applications. Paper read at Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA '09., 22-25 Sept. 2009.
- DetikNews. 2012 "Sidak Pencurian Listrik, PLN Gandeng Polisi Periksa 2 Pedagang." *Selasa, 24/04/2012*
- Douglas, A., Hoang-Anh Fidaleo, and Mohan Nguyen, Trivedi. 2004. The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data
-

- 
- in video-sensor networks. In *Proceedings of the ACM 2nd international workshop on Video surveillance & amp ; sensor networks*. New York, NY, USA: ACM.
- Durresi, A. Durresi, M. Barolli, L. 2008. Security of Mobile and Heterogeneous Wireless Networks in Battlefields. Paper read at Parallel Processing - Workshops, 2008. ICPP-W '08. International Conference on.
- Elizabeth, A. Basha, Ravela Sai, and Rus Daniela. 2008. Model-based monitoring for early warning flood detection. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. Raleigh, NC, USA: ACM.
- Elkamchouchi, H. M., A. A. M. Emarah, and E. A. A. Hagra. 2006. A New Secure Hash Dynamic Structure Algorithm (SHDSA) for Public Key Digital Signature Schemes. Paper read at Radio Science Conference, 2006. NRSC 2006. Proceedings of the Twenty Third National, 14-16 March 2006.
- Fei, Hu. , and Kumar. Sunil. 2003. "QoS considerations in wireless sensor networks for telemedicine " In *Proceedings of SPIE ITCOM Conference, Orlando, FL*
- Fengchao, Chen RongLin, Li. 2011. Single sink node placement strategy in wireless sensor networks. Paper read at Electric Information and Control Engineering (ICEICE), 2011 International Conference on, 15-17 April 2011.
- García Villalba, Luis Javier, and Ana Lucila Sandoval Orozco, Triviño Cabrera, Alicia ,Barencó Abbas, Cláudia Jacy. 2009. "Routing protocols in wireless sensor networks." *Sensors* no. 9 (11):8399-8421.
- Gilman, Tolle, Polastre Joseph, Robert, Szewczyk, David, Culler,Neil, Turner, and Tu Kevin, Stephen, Burgess,Todd, Dawson,Phil, Buonadonna,David, Gay,Wei, Hong. 2005. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*. San Diego, California, USA: ACM.
- Grieco, Luigi Alfredo, Gennaro Boggia, and Sabrina Sicari, Colombo, Pietro. 2009. Secure Wireless Multimedia Sensor Networks: A Survey. Paper read at Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBICOMM '09. Third International Conference on.
- Gungor, V. C., and G. P. Hancke. 2009. "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches." *Industrial Electronics, IEEE Transactions on* no. 56 (10):4258-4265.
- Han, Song, Elizabeth Chang, and Li Gao, Dillon, Tharam. 2006. "Taxonomy of attacks on wireless sensor networks." In *EC2ND 2005*, 97-105. Springer.
- Hani, Alzaid, Foo Ernest, and Nieto Juan Gonzalez. 2008. Secure data aggregation in wireless sensor network: a survey. In *Proceedings of the sixth Australasian conference on*
-

- 
- Information security - Volume 81*. Wollongong, NSW, Australia: Australian Computer Society, Inc.
- HarianjoglosemarNews. 2012 "pencurian-listrik-pln-solo-tanggung-rp-21-m."
- Harjito, Bambang 2003. "Watermarking Technique based on Linear Feed Back Shift Register (LFSR), ." *Seminar Nasional Konferda ke –9 Himpunan Matematika Wilayah Jateng dan DIY di FMIPA UNS*
- Healy, M., T. Newe, and E. Lewis. 2008. Wireless Sensor Node hardware: A review. Paper read at Sensors, 2008 IEEE, 26-29 Oct. 2008.
- Honggang Wang, Dongming Peng , and Hamid Sharif and Hsiao-Hwa Chen Wei Wang. 2008. Energy-Aware Adaptive Watermarking for Real-Time Image Delivery in Wireless Sensor Networks. Paper read at Communications, 2008. ICC '08. IEEE International Conference on.
- I. Cox, M. Miller, and J. Bloom. 2002. "Digital Watermarking." *Morgan Kaufmann*,.
- Jessica, Fang. , and Miodrag. Potkonjak 2003. " Real-time watermarking techniques for sensor networks " *Proceedings-SPIE The International Society for optical Engineering (ISSU 5020):391-402* doi: 10.1117/12.479736
- Jian, Liu, and He Xiangjian. 2005. A Review Study on Digital Watermarking. Paper read at Information and Communication Technologies, 2005. ICICT 2005. First International Conference on, 27-28 Aug. 2005.
- Juma, H., I. Kamel, and L. Kaya. 2008. Watermarking sensor data for protecting the integrity. Paper read at Innovations in Information Technology, 2008. IIT 2008. International Conference on.
- Kahn, JM, RH Katz, and KSJ Pister. 1999. "Next Century Challenges: Mobile Networking for "Smart Dust"." *In Proceedings of the ACM MobiCom '99, Washington, DC, USA*,.
- Kamel, I. 2011. "A Lightweight Data Integrity Scheme for Sensor Networks." *Sensors* no. 11 (4):4118.
- Kamel, Ibrahim, and Hussam Juma. 2011. "A lightweight data integrity scheme for sensor networks." *Sensors* no. 11 (4):4118-4136.
- Karlof, C. 2003. "Secure routing in wireless sensor networks: Attacks and countermeasures." *Ad Hoc Networks* no. 1 (2-3):293.
- Karlof, Chris, Naveen Sastry, and David Wagner. 2004. TinySec: a link layer security architecture for wireless sensor networks. Paper read at Proceedings of the 2nd international conference on Embedded networked sensor systems.
-

- 
- Kaur, S. 2010 *Digital Watermarking of ECG Data for Secure Wireless Communication International Conference on Recent Trends in Information, Telecommunication and Computing*.
- Koushanfar, Farinaz., and Miodrag. Potkonjak. 2007. "Watermarking Technique for Sensor Networks: Foundations and Applications." *Book chapter, in 'Security in Sensor Networks', Yang Xiao*
- Kundur, D, Ndili Unoma, Okorafor, , and Luh William. 2006. HoLiSTiC: Heterogeneous Lightweight Sensornets for Trusted Visual Computing. Paper read at Intelligent Information Hiding and Multimedia Signal Processing, 2006. IHH-MSP '06. International Conference on, Dec. 2006.
- Leigh, Jason, Luc Renambot, and Andrew Johnson, Jeong, Byungil, Jagodic, Ratko, Schwarz, Nicholas, Svistula, Dmitry, Singh, Rajvikram, Aguilera, Julieta, Wang, Xi. 2006. "The global lambda visualization facility: an international ultra-high-definition wide-area visualization collaboratory." *Future Generation Computer Systems* no. 22 (8):964-971.
- Li, Ming, and Paul Vitanyi. 2008. *An Introduction to Kolmogorov Complexity and Its Applications*. Vol. Third Edition New York: Springer Verlag.
- Lian, Shiguo, Dimitris Kanellopoulos, and Giancarlo Ruffo. 2009. "Recent advances in multimedia information system security." *Informatica* no. 33 (1):3-24.
- Licks, V., and R. Jordan. 2005. "Geometric attacks on image watermarking systems." *MultiMedia, IEEE* no. 12 (3):68-78. doi: 10.1109/MMUL.2005.46.
- Lin, K. W., Ming-Hua, and V. S. Hsieh Tseng. 2009. Mining Temporal Region-Based Service Patterns for Cooperative Caching in Wireless Multimedia Sensor Networks. Paper read at Intelligent Information Hiding and Multimedia Signal Processing, 2009. IHH-MSP '09. Fifth International Conference on, 12-14 Sept. 2009.
- Lingxuan, Hu, and D. Evans. 2003. Secure aggregation for wireless networks. Paper read at Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on.
- Lorincz, Konrad, David J Malan, and Thaddeus RF Fulford-Jones, Nawoj, Alan, Clavel, Antony, Shnayder, Victor, Mainland, Geoffrey, Welsh, Matt, Moulton, Steve. 2004. "Sensor networks for emergency response: challenges and opportunities." *Pervasive Computing, IEEE* no. 3 (4):16-23.
- Manel Guerrero-Zapata, Ruken Zilan , Jos´e M. Barcel´o-Ordinas, Kemal Bicakci, Bulent Tavli. 2009. "The Future of Security in Wireless Multimedia Sensor Networks."
- Masood, H., U. Haider, and Rehman Sadiq ur, Khosa, I. 2010. Secure communication in WMSN. Paper read at Information Networking and Automation (ICINA), 2010 International Conference on, 18-19 Oct. 2010.
-

- 
- Massey, James L. 1992. *Contemporary cryptology: an introduction*: Contemporary Cryptology: The Science of Information Integrity, GJ Simmons, ed., IEEE Press.
- Mauricio, Capra Milena, Benford Leif Radenkovic Steve, and Drozd Martin Oppermann Adam, Flintham. 2005. The multimedia challenges raised by pervasive games. In *Proceedings of the 13th annual ACM international conference on Multimedia*. Hilton, Singapore: ACM.
- Meingast, M., T. Roosta, and S. Sastry. 2006. Security and Privacy Issues with Health Care Information Technology. Paper read at Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE.
- Ming-Kuei, Hu. 1962. "Visual pattern recognition by moment invariants." *Information Theory, IRE Transactions on* no. 8 (2):179-187.
- Noury, Norbert . Hervé, Thierry, Rialle, Vicent, Virone, Gilles, Eric. Morey Mercier, Gilles, Moro, Aldo, and Thierry Porcheron. 2000. Monitoring behavior in home using a smart fall sensor and position sensors. Paper read at Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000.
- NRI, Net Resources International. 2012. "Kansas City Scout, United States of America <http://www.roadtraffic-technology.com/projects/kansas/>."
- Padmavathi, G. , D. Shanmugapriya, and M. Kalaivani. 2010. Digital watermarking technique in vehicle identification using wireless sensor Networks. Paper read at Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, 20-22 Aug. 2010.
- Padmavathi, G., D. Shanmugapriya, and M. Kalaivani. Digital watermarking technique in vehicle identification using wireless sensor Networks. Paper read at Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, 20-22 Aug. 2010.
- Pathan, A. S. K., Lee. Hyung-Woo, and Hong. Choong Seon. 2006. Security in wireless sensor networks: issues and challenges. Paper read at Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, 20-22 Feb. 2006.
- Pingping, Yu Suying, Xu Yu Yao Jiangtao, and Chang Zhang Ye. 2009. Copyright Protection for Digital Image in Wireless Sensor Network. Paper read at Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on.
- Pister, Kristofer. S. J. 2003. Smart dust-hardware limits to wireless sensor networks. Paper read at Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on, 19-22 May 2003.
- Potdar, V. M., S. Han, and E. Chang. 2005. A survey of digital image watermarking techniques. Paper read at Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on, 10-12 Aug. 2005.
-

- 
- Potdar, V., A. Sharif, and E. Chang. 2011. *The Industrial Electronics Handbook, Second Edition*. Edited by In: B. M. Wilamowski & J. D. Irwin eds: Boca Raton, FL, USA: CRC Press. Ch. 11.
- Potdar, V., A. Sharif, and E. Chang. 2009. Wireless Sensor Networks: A Survey. Paper read at Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on, 26-29 May 2009.
- Radu, Sion, Atallah Mikhail, and Prabhakar Sunil. 2004. Resilient rights protection for sensor streams. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*. Toronto, Canada: VLDB Endowment.
- Raghu, K. Ganti Praveen, Jayachandran Haiyun, and F. Abdelzaher Luo Tarek. 2006. Datalink streaming in wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*. Boulder, Colorado, USA: ACM.
- Reeves, A.A 2005. "Remote monitoring of patients suffering from early symptoms of dementia." in *Proc. Int. Workshop Wearable Implantable Body Sensor Networks*:London, U.K., Apr. 2005.
- Ren, Baowei Wang ; Xingming Sun ; Zhiqiang Ruan ; Heng. 2011. "Multi-mark: Multiple Watermarking Method for Privacy Data Protection in Wireless Sensor Networks." *Information Technology Journal* no. 10 (4):833-840. doi: 10.3923/itj.2011.833.840
- Ren, Xiuli, and Haibin Yu. 2006. "Security mechanisms for wireless sensor networks." *IJCSNS International Journal of Computer Science and Network Security* no. 6 (3):155-156.
- Rong, Xiao, Sun Xingming, and Yang Ying. 2008. Copyright Protection in Wireless Sensor Networks by Watermarking. Paper read at Intelligent Information Hiding and Multimedia Signal Processing, 2008. IJHMSP '08 International Conference on.
- Roosta, Tanya, Shiuhyng Shieh, and Shankar Sastry. 2006. Taxonomy of security attacks in sensor networks and countermeasures. Paper read at The First IEEE International Conference on System Integration and Reliability Improvements.
- Ruiz, L. B., J. M. Nogueira, and A. A. F. Loureiro. 2003. "MANNA: a management architecture for wireless sensor networks." *Communications Magazine, IEEE* no. 41 (2):116-125. doi: 10.1109/mcom.2003.1179560.
- Sangeon, Park Navrati, Saxena Jitae, Shin Minsoo, Suk. 2008. An energy-efficient and QoS-based MAC layer protocol for WMSN. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. Suwon, Korea: ACM.
- Sarma, Hiren Kumar Deva, and Avijit Kar. 2006. Security threats in wireless sensor networks. Paper read at Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International.
-

- 
- Sharif, A., V. Potdar, and E. Chang. 2009. Wireless multimedia sensor network technology: A survey. Paper read at Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on.
- Sherekar, S. 2011. "Attacks and Countermeasures on Digital Watermarks: Classification, Implications, Benchmarks." *International journal of computer science and applications* no. 4 (2).
- Subash, T. D., and C. Divya. 2011. Double hash function scheme in wireless sensor networks. Paper read at Information and Communication Technologies (WICT), 2011 World Congress on, 11-14 Dec. 2011.
- T.M. Salvatore, F.S. Gerald 1995. "Design and Natural Science Research on Information Technology." *Decision Support System* no. Vol. 15:251-266, 1995.
- Tahir, Hasan, and S Shah. 2008. Wireless sensor networks-a security perspective. Paper read at Multitopic Conference, 2008. INMIC 2008. IEEE International.
- Tao, Chen, Wang Jingchun, and Zhou Yonglei. 2001. Combined digital signature and digital watermark scheme for image authentication. Paper read at Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on, 2001.
- Tatiana Bokareva, Wen Hu, Salil Kanhere, Branko Ristic, Neil Gordon, Travis Bessell, Mark Rutten , sanjay Jha. 2006. "Wireless Sensor Networks for Battlefield Surveillance." *Land Warfare Conference, Brisbane October 2006*.
- Vasilescu, I., K. Kotay, and M. Dunbabin D. Rus, P. Corke. 2005. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*. San Diego, California, USA: ACM.
- Vivas, José L., Carmen Lopez Fernández-Gago, Javier, , and Andrés Benjumea. 2010. "A security framework for a workflow-based grid development platform." *Computer Standards & Interfaces* no. In Press, Corrected Proof.
- Voloshynovskiy, S. Pereira, S., T. Pun, and J. J. Su Eggers, J. K. 2001. "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks." *Communications Magazine, IEEE* no. 39 (8):118-126. doi: 10.1109/35.940053.
- W Wang, D Peng, H Wang, Y Yang, H Sharif 2008. "Position Based Unequal Error Protection for Image Transmission with Energy Constraint over Multirate XPD MIMO Sensor Networks." *IEE GLOBECOM" 2008 proceedings*.
- Wang, Baowei., Xingming. Sun, and Zhiqiang Ruan, Ren, Heng. 2011. "Multi-mark: Multiple Watermarking Method for Privacy Data Protection in Wireless Sensor Networks." *Information Technology Journal* no. 10 (4):833-840.
-



- 
- Wang, Honggang. 2010. "Communication-resource-aware adaptive watermarking for multimedia authentication in wireless multimedia sensor networks." *The Journal of Supercomputing*:1-15. doi: 10.1007/s11227-010-0500-5.
- Wang, Xiang-Yang, Zi-Han Xu, and Hong-Ying Yang. 2009. "A robust image watermarking algorithm using SVR detection." *Expert Systems with Applications* no. 36 (5):9056-9064.
- Warneke, B. Last, M.Liebowitz, B.Pister, K. S. J. 2001. "Smart Dust: communicating with a cubic-millimeter computer." *Computer* no. 34 (1):44-51.
- Wen-bo, Zhang, Xu Hai-feng, and Sun Pei-gen. 2010. A Network Management Architecture in Wireless Sensor Network. Paper read at Communications and Mobile Computing (CMC), 2010 International Conference on, 12-14 April 2010.
- Wenjing, Lou, and Kwon Younggoo. 2006. "H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks." *Vehicular Technology, IEEE Transactions on* no. 55 (4):1320-1330.
- Wenjun, Zeng, and B. Liu. 1999. "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images." *Image Processing, IEEE Transactions on* no. 8 (11):1534-1548.
- Werner-Allen, G., K. Lorincz, and M. Marcillo Ruiz, O.Johnson, J.Lees, J.Welsh, M. 2006. "Deploying a wireless sensor network on an active volcano." *Internet Computing, IEEE* no. 10 (2):18-25.
- Wuyungerile, Li, M. Bandai, and T. Watanabe. 2010. Tradeoffs among Delay, Energy and Accuracy of Partial Data Aggregation in Wireless Sensor Networks. Paper read at Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 20-23 April 2010.
- Xiangjun, Zhu, Ying Shaodong, and Ling Le. 2008. Multimedia sensor networks design for smart home surveillance. Paper read at Control and Decision Conference, 2008. CCDC 2008. Chinese, 2-4 July 2008.
- Xiangqian, C, Makki, K. Kang, Yen Pissinou, N. 2009. "Sensor network security: a survey." *Communications Surveys & Tutorials, IEEE* no. 11 (2):52-73.
- Xiao, Yang. 2006. "Wireless Sensor Network Security: A Survey." *Security in Distributed, Grid, and Pervasive Computing 2006 Auerbach Publications, CRC Press*.
- Xiaomei, Dong , and Li Xiaohua. 2009. An Authentication Method for Self Nodes Based on Watermarking in Wireless Sensor Networks. Paper read at Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on, 24-26 Sept. 2009.
-

- 
- Xiaomei, Dong Xiaohua, Li. 2009. An Authentication Method for Self Nodes Based on Watermarking in Wireless Sensor Networks. Paper read at Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on, 24-26 Sept. 2009.
- Xing, Kai, Shyaam Sundhar Rajamadam Srinivasan, Jose, Major, and Jiang. Cheng Li, Xiuzhen. 2010. "Attacks and countermeasures in sensor networks: a survey." In *Network Security*, 251-272. Springer.
- Xuejun, R. 2010. " A sensitivity data communication protocol for WSN based on digital watermarking " *School of Information and Technology, Northwestern University, Xi'an 710127, China*.
- Yick, Jennifer, Biswanath Mukherjee, and Dipak Ghosal. 2008. "Wireless sensor network survey." *Computer Networks* no. 52 (12):2292-2330.
- Zhang, Wei., Yonghe. Liu, and Sajal K. De Das, Pradip. 2008. "Secure data aggregation in wireless sensor networks: a watermark based authentication supportive approach." *Pervasive and Mobile Computing* no. 4 (5):658-680.
-

# APPENDIX I

## APPENDIX I A : LKR Watermarking Technique Matlab Code

### 1. *menuLKR.m*

```

lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' MAIN MENU GENERATE WATERMARK ')
disp(' LKR WATERMARKING TECHNIQUE ')
disp('=====')
disp('1. SetUpnetworksCopyrightWSNs ')
disp('2. GenerateCoverMedium ')
disp('3. Generate stream watermark ')
disp('4. Cover medium with data message ')
disp('5. Cover medium with data message with constraints ')
disp('6. Extract ')
disp('7. Attack_data_False_Insertion and detect ')
disp('6. Attack_data_Deletion and detect')
disp('8. Attack_data_replication aand detect')
disp('9. Attack_data_Sybil and detect')
disp('10. Menu ')
disp('11.Exit ')
disp('=====')
 pilih = input ('Which one do you want = ');
if pilih==1
 SetUpnetworksCopyrightWSNs
elseif pilih==2
 GenerateCoverMedium
elseif pilih==3
 Watermarkgeneration
elseif pilih==4
 EmbeddingMessageD
elseif pilih==5
 EmbeddingMsdWConstraints
elseif pilih==6
 Extract
elseif pilih==7
 DetectAttackFalseInsert
elseif pilih==8
 DetectAttackDataDeletion
 elseif pilih==9
 DetectAttackDataReplication
elseif pilih==10
 DetectAttackDataSybill
elseif pilih==11
 MenuLKR
else
 exit
end
lagi=input('Anda ingin mengulangi lagi ? (Y/N)', 's')
end
disp('')

```

---

```
disp('Copyright protection of data in WSN using LKR watermarking ')
disp('')
```

---

## 2. *SetUPnetworksCopyrightWSNs.m*

```
lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' LKR WATERMARKING TECHNIQUE ')
disp(' Copyright protection of scalar data in WSNs ')
disp(' NETWORK SET UP ')
disp('=====')
d=[];
noOfNodes = 75;
rand('state', 0);
figure(1);
clf;
hold on;
L = 500;
R = 100; % maximum range;
netXloc = rand(1,noOfNodes)*L;
netYloc = rand(1,noOfNodes)*L;

Coordinate1 =[];
for i = 1:noOfNodes
plot(netXloc(i), netYloc(i), 'o',...
 'MarkerEdgeColor','k',...
 'MarkerFaceColor','r',...
 'MarkerSize',10);
xlabel(' Coordinate of X axis')
ylabel(' Coordinate of Y absis ')
title('EXPERIMENT SET UP COPYRIGHT PROTECTION IN WIRELESS SENSOR
NETWORKs')
text(netXloc(i), netYloc(i), num2str(i));

for j = 1:noOfNodes
distance = sqrt((netXloc(i) - netXloc(j))^2 + (netYloc(i) -
netYloc(j))^2);
Coordinate1(i,1)= netXloc(i)
Coordinate1(j,2)= netYloc(j)
disp(sprintf('%.4g | %.4g | %.4g ',i, netXloc(i), netYloc(j)));
if distance <= R
matrix(i, j) = 1; % there is a link;
line([netXloc(i) netXloc(j)], [netYloc(i) netYloc(j)], 'LineStyle', ':');
else
matrix(i, j) = inf;
end;
end;
end;
xlswrite('Coordinate1.xls',Coordinate1)
s1=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\Coordinate1.xls',1,'A4:B75');
s2=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\Coordinate1.xls',1,'A1:B3');
Coordinate2= [s1 ; s2]
xlswrite('CoordinateWSN1.xls',Coordinate2)
```

---

---

```

s3=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\Coordinate1.xls',1,'A7:B75');
s4=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\Coordinate1.xls',1,'A1:B6');
Coordinate3= [s3 ; s4]
xlswrite('CoordinateWSN2.xls',Coordinate3)

DATAposisi = [Coordinate1 Coordinate2 Coordinate3];
xlswrite('threeOtherNodesWMSNs.xls', DATAposisi);

disp('Generate time arrival and departure sensor ')
time=[];
for j = 1:noOfNodes

 disp('-----')
 disp('No | tDA | tDB | tDC ')
 disp('-----')
 for i=1:noOfNodes
 disp(sprintf('%.4g | %.4g | %.4g | %.4g |',i, random('unif',0,1) ,
random('unif',0,1) , random('unif',0,1)));
 time(i,1)= random('unif',0,1);
 time(i,2)= random('unif',0,1);
 time(i,3)= random('unif',0,1);
 end
end
disp('Generate Temperature of the propagation media ')
xlswrite('time123.xls',time)
temp=[];
Max_Temp=noOfNodes;
temp=rand(Max_Temp,1)*Max_Temp
xlswrite('temp.xls',temp)

disp(' Generate the feasibility of value ')
fisible=[];
for j = 1:noOfNodes
 disp('-----')
 disp('No | thou1 | thou2 | thou3 | thou4 ')
 disp('-----')
 for i=1:noOfNodes
 disp(sprintf('%.4g | %.4g | %.4g | %.4g |%.4g ',i,
random('unif',0,1) , random('unif',0,1) , random('unif',0,1),
random('unif',0,1)));
 fisible(i,1)=random('unif',0,1) ;
 fisible(i,2)=random('unif',0,1) ;
 fisible(i,3)=random('unif',0,1) ;
 fisible(i,4)=random('unif',0,1) ;
 end
end
xlswrite('Fisible1234.xls',fisible)
DATAlengkap = [DATAposisi time temp fisible];
xlswrite('datalengkapWSNs.xls',DATAlengkap);
A=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\datalengkapWSNs.xls',1,'A1:Q32');
xlswrite('data32nodesWSNs.xls',A)
lagi=input('Do you want to return ? (Y/N)','s')
end
disp('')
disp('Copyright protection of data in WSN using LKR watermarking ')

```

---

```
disp('')
MenuLKR
```

### 3. Watermarkgeneration.m

```
clc
lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' MAIN MENU GENERATE WATERMARK ')
disp(' LKR WATERMARKING TECHNIQUE ')
disp(' Copyright protection of scalar data in WSNs ')
disp(' Gathering data by WSNs into binary sequence ')
disp(' Generating Watermark stream using LFSR ')
disp(' Watermark Constraints using Kolmogorov rule ')
disp(' Generating Message Sensed Data ')
disp('-----')
pilih = input ('Which one do you want = ');
if pilih==1
disp('-----')
disp(' Capturing data by WSNs into binary sequence ')
disp(' Converting the data sensory decimal to binary sequence ')
disp('-----')
sensornode=input('Capture scalar data node will be protected =');
ori=de2bi(sensornode);
[m1 n1]=size(ori);
oriseq=[];
for i=1:m1
for j=1:n1
oriseq(1,j)=ori(1,j);
end
end
s=fliplr(oriseq)
%elseif pilih==2
disp('-----')
disp(' Initial state of LFSR from converting binary sequece ')
disp(' Watermark Key is a tap position ')
disp('-----')
%s= input ('Implement sensor data to formed such as [1 0 ... 1] =');
t = input('Watermak Key is like tap position of f(x) is =');
n=length(s);
c(1,:)=s';
m=length(t);
for k=1:2^n-2;
b(1)=xor(s(t(1)), s(t(2)));
if m>2;
for i=1:m-2;
b(i+1)=xor(s(t(i+2)), b(i));
end
end
j=1:n-1;
s(n+1-j)=s(n-j);
s(1)=b(m-1);
c(k+1,:)=s;
end
seq=c(:,n)';
%A = seq;
```

---

```

 [row col]=size(seq);
 disp('cutting the infinity binary stream')
 A = seq(1,1:28);
 [row1 coll]=size(A);
 disp('Length of column cutting the binary = '); fprintf('%6.4 \t',coll)
 st=[];
 di= input ('How many rows do you want create for dividing binary = ');
 for k=1:4
 st(k,:)=A(1, ((k-1)*coll/di)+1:(k *coll)/di);
 end
 st
 %disp('Save to xls')
 xlswrite('Signal.xls',st)
elseif pilih==3

disp('-----')
disp(' Watermark dividing using Kolmogorov rule ')
disp('-----')
kolomogorof
x='et'; y='eda'; z='edb';
r='edc'; s='dl'; t='d2'; u='d3';
disp('The watermark constrains of =');
[m n]=size(st);
ff=[];
ff=fopen('gen.txt','w');
for i=1:m;
 fprintf(' \n');
for j=1:n;
if st(i,j)==1;
 if j==1;
 fprintf(ff,'%6.3s \t',x);
 elseif j==2;
 fprintf(ff,'%6.3s \t',y);
 elseif j==3;
 fprintf(ff,'%6.3s \t',z);
 elseif j==4;
 fprintf(ff,'%6.3s \t',r);
 elseif j==5;
 fprintf(ff,'%6.3s \t',s);
 elseif j==6;
 fprintf(ff,'%6.3s \t',t);
 else
 fprintf(ff,'%6.3s \t',u);
 end
else st(i,j)==0;
 fprintf(ff,'%6.3s \t','0');
end
end
disp('');
end
fclose(ff);
fid1 = fopen('gen.txt');
tline = fgetl(fid1);
while ischar(tline);
 disp(tline);
 tline = fgetl(fid1);
end
de=textread('gen.txt','%s');
```

---

---

```

deA=sym([de']);
[row2 col2]=size(deA);
t1= input ('How many rows do you want create watermark constraints = ');
WC=reshape(deA,t1,[]);
disp('Watermark Constraints are ')
for i=1:4
 disp(sum(WC(i,:)))
end
[m n]=size(WC);
ffw=[];
ffw=fopen('WaCons.txt','w');
for i=1:m;
 fprintf(' \n')
for j=1:n;
if st(i,j)~=1;
 if j==1;
 fprintf(ffw,'%6.3s \t',x);
 elseif j==2;
 fprintf(ffw,'%6.3s \t',y);
 elseif j==3;
 fprintf(ffw,'%6.3s \t',z);
 elseif j==4;
 fprintf(ffw,'%6.3s \t',r);
 elseif j==5;
 fprintf(ffw,'%6.3s \t',s);
 elseif j==6;
 fprintf(ffw,'%6.3s \t',t);
 else
 fprintf(ffw,'%6.3s \t',u);
 end
else st(i,j)==0;
 fprintf(ffw,'%6.3s \t','0');
end
end
disp('');
end
fclose(ffw);
else
 MenuLKR
end
lagi=input('Do you want to repeat ? (Y/N)','s')
end

```

---

#### 4. DetectAttackFalseInsert.m

```

clc
lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' LKR WATERMARKING TECHNIQUE ')
disp(' COPYRIGHT OF SCALAR DATA IN WSNs ')
disp(' 1. DETECT FALSE DATA INSERTION ')
disp(' 2. Menu LKR Watermarking technique ')
disp('=====')
pilih=input('What Do you want = ');

```

---



---

```

if pilih==1
disp(' Read Message Sensed Data(MSD) is = ');
%Msd= xlsread('C:\Users\Bambang\Desktop\Gambar Thesis Bambang\Program
Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\MSD.xls',1,'A1:G7');
Msd= xlsread(
'C:\Users\Bambang\Dropbox\Photos\CopyrightprotectionOfWSNs\AllAttack\XLS\
MSD.xls',1,'A1:G7');
disp(Msd)
msd1=Msd(1,1);msd2=Msd(1,2);msd3=Msd(1,3);
msd4=Msd(1,4);msd5=Msd(1,5);msd6=Msd(1,6); msd7=Msd(1,7);
N1=[]; N2=[];N3=[];N4=[];N5=[];N6=[]; N7=[]; f1=[];
%A=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\CopyrightprotectionOfWSNs\data32nodesWSNs.xls',1,'A1:Q32');
%A=xlsread ('C:\tomlab\data32nodesWSNs.xls',1,'A1:Q32');
%A=xlsread('C:\Users\Bambang\Desktop\Gambar Thesis Bambang\Program Matlab
untuk secure data
WSNs\CopyrightprotectionOfWSNs\data32nodesWSNs.xls',1,'A1:Q32');
A=xlsread('C:\Users\Bambang\Dropbox\Photos\CopyrightprotectionOfWSNs\Alla
tack\XLS\data32nodesWSNs.xls',1,'A1:N32');
%C:\Users\Bambang\Dropbox\Photos\CopyrightprotectionOfWSNs\AllAttack

n=input('How many experiment what do you want = ');
for i=1:n
xa = A(i,1);xb =A(i,3);xc = A(i,5);ya = A(i,2);yb = A(i,4);yc = A(i,6);Tc
= A(i,10);
tda = A(i,7);tdb = A(i,8);
tdc = A(i,9);thou1 = A(i,11); thou2 = A(i,12); thou3 = A(i,13);thou4 =
A(i,14);

toms xd yd et eda edb edc d1 d2 d3
f =
msd1*abs(et)+msd2*abs(eda)+msd3*abs(edb)+msd4*abs(edc)+msd5*d1+msd6*d2+ms
d7*d3; %with data message
%f = 1*abs(et)+14*abs(eda)+0*abs(edb)+13*abs(edc)+12*d1+12*d2+7*d3;
%with data message
c = {
-d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))*(tda+eda) <= d1
-d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))*(tdb+edb) <= d2
-d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))*(tdc+edc) <= d2
0<=d1
0<=d2
0<=d3 };
solution = ezsolve(f,c);
N1(i)=subs([et] , solution);
N2(i)=subs([eda], solution);
N3(i)=subs([edb], solution);
N4(i)=subs([edc], solution);
N5(i)=subs([d1], solution);
N6(i)=subs([d2], solution);
N7(i)=subs([d3], solution);
f1(i)=subs([f], solution);
end
PreResultA=[N1 ; N2; N3; N4; N5; N6; N7];
fobj1 = [f1];
disp(' et eda edb edc d1 d2 d3 ');
ResultsA = PreResultA';
disp(ResultsA);

```

---

---

```

disp(fobj1');
xlswrite('ResultsWithoutwater.xls',ResultsA);

% Stop for only message data=====

for i=1:n
 xa = A(i,1);xb =A(i,3);xc = A(i,5);ya = A(i,2);yb = A(i,4);yc =
A(i,6);Tc = A(i,10);
 tda = A(i,7);tdb = A(i,8);
 tdc = A(i,9);thou1 = A(i,11); thou2 = A(i,12); thou3 = A(i,13);thou4 =
A(i,14);
 toms xd yd et eda edb edc d1 d2 d3
 f =
msd1*abs(et)+msd2*abs(eda)+msd3*abs(edb)+msd4*abs(edc)+msd5*d1+msd6*d2+ms
d7*d3; %with data message
 %f = 1*abs(et)+14*abs(eda)+0*abs(edb)+13*abs(edc)+12*d1+12*d2+7*d3;
 %with data message
 c = {
 -d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))*(tda+eda) <= d1
 -d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))*(tdb+edb) <= d2
 -d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))*(tdc+edc) <= d2
 0<= edb+edc+d1+d3 <=thou1 % wateramrk constraints
 0<= eda+edc+d1+d2 <= thou2 % wateramrk constraints
 0<= et+edb+d1+d2 <= thou3 % wateramrk constraints
 0<= eda+edb+edc+d1+d3 <= thou4 % wateramrk constraints
 0<=d1
 0<=d2
 0<=d3 };
 solution = ezsolve(f,c);
 N1(i)=subs([et] , solution);
 N2(i)=subs([eda], solution);
 N3(i)=subs([edb], solution);
 N4(i)=subs([edc], solution);
 N5(i)=subs([d1], solution);
 N6(i)=subs([d2], solution);
 N7(i)=subs([d3], solution);
 f1(i)=subs([f], solution);
end
PreResultB=[N1 ; N2; N3; N4; N5; N6; N7];
fobj1 = [f1];
disp(' et eda edb edc d1 d2 d3 ');
ResultsB = PreResultB';
disp(ResultsB);
disp(fobj1');
xlswrite('ResultsWithWatreamrk.xls',ResultsB);

for i=1:n
 xa = A(i,1);xb =A(i,3);xc = A(i,5);ya = A(i,2);yb = A(i,4);yc =
A(i,6);Tc = A(i,10);
 tda = A(i,7);tdb = A(i,8);
 tdc = A(i,9);thou1 = A(i,11); thou2 = A(i,12); thou3 = A(i,13);thou4 =
A(i,14);
 toms xd yd et eda edb edc d1 d2 d3
 f =
msd1*abs(et)+msd2*abs(eda)+msd3*abs(edb)+msd4*abs(edc)+msd5*d1+msd6*d2+ms
d7*d3; %with data message
 %f = 1*abs(et)+14*abs(eda)+0*abs(edb)+13*abs(edc)+12*d1+12*d2+7*d3;
 %with data message

```

---

---

```

c = {
-d1<= sqrt((xd-xa)^2+(yd+ya)^2)-(331.4+0.6*(Tc+et))*(tda+eda) <= d1
-d2<= sqrt((xd-xb)^2+(yd-yb)^2)-(331.4+0.6*(Tc+et))*(tdb+edb) <= d2
-d3<= sqrt((xd-xc)^2+(yd-yc)^2)-(331.4+0.6*(Tc+et))*(tdc+edc) <= d2
 0<= edb+edc+d1+d3 <=thou1 % wateramrk constraints
 %0<= d2+d3 <= thou2 % wateramrk constraints
 %0<= eda+edb+edc+d3 <= thou3 % wateramrk constraints
 %0<= et+d1+d2+d3 <= thou4 % wateramrk constraints
 0<=eda+edc+d2+d3<=thou2 % False watermark Constraint
 0<=eda+d1+d3<=thou3 % False watermark Constraint
 0<=eda+edb+edc+d2+d3<=thou4 % False watermark Constraint

 0<= d1+d2+d3 <= thou2 % wateramrk constraints
 0<= eda+edc++d1+d3 <= thou3 % wateramrk constraints
 0<= et+d2 <= thou4 % wateramrk constraints
 0<=d1
 0<=d2
 0<=d3
};
solution = ezsolve(f,c);
N1(i)=subs([et] , solution);
N2(i)=subs([eda], solution);
N3(i)=subs([edb], solution);
N4(i)=subs([edc], solution);
N5(i)=subs([d1], solution);
N6(i)=subs([d2], solution);
N7(i)=subs([d3], solution);
f1(i)=subs([f], solution);
end
PreResultC=[N1 ; N2; N3; N4; N5; N6; N7];
fobj1 = [f1];
disp(' et eda edb edc d1 d2 d3
');
ResultsC = PreResultC';
disp(ResultsC)
disp(fobj1')
xlswrite('ResultsFalsecoy3.xls',ResultsC);

% To verify the presence of the watermark, the similarity between the
normalized difference error from
% the optimal solution between the watermarked solution and the solution
obtained without watermarked
% The similarity measure is given by the normalized correlation
coefficient
% Computing the similarity between the normalized difference error from
the optimal solution between
% the watermarked solution X' and the solution obtained without
watermarked X so C = X'-X
% the measure using sim(C,X')= (C*X') /sqrt(X'*X') as a threshold

disp('1. Cover medium with data message ')
disp('The solutions are ')
disp(ResultsA)
disp('2. Embedding Cover medium data message and watermark constraint')
disp('The solutions are ')
disp(ResultsB)
disp('3. Attack data False Insertion ')
disp('The solutions are ')

```

---

---

```

disp(ResultsC)
disp (' 4. Detecting attack data False Insertion using similarity ')
ALL_Data_solution = [ResultsA ; ResultsB ; ResultsC];

disp('The normalized difference error from the optimal solution between
the watermarked solution X')
disp('and the solution obtained without watermarked X')
disp('Threshold is the value that is used to determine the watermark
constraints changes or not ')

%C = ResultsB - ResultsA ;
%threshold = (C.* ResultsB)/sqrt(ResultsB .*ResultsB)
%disp('The value of threshold is = '); disp(threshold)

%disp('-----')
%disp('The similarity use sim(C,X) = (C*X) /sqrt(X*X) ')
%C1 = ResultsC-ResultsA;
%similarity = (C1.*ResultsC)/sqrt(ResultsC .*ResultsC)
%CheckRobusnessInsert
ALATCHECK
disp('The value of similarity is ');
disp(similarity)
disp('To check whether the watermark cosntraints changes or not ')

 if threshold >= similarity
 disp(' The watermark constraints do not change and then check all of
constraints ')
 disp('-----')
 disp(' 1. Check watermark constraints')
 et=ResultsA(1,1); eda=ResultsA(1,2);edb=ResultsA(1,3);
edc=ResultsA(1,4);
 d1=ResultsA(1,5);d2=ResultsA(1,6);d3=ResultsA(1,7);
 if ((edc+d1+d2+d3 <=thou1) && (d2+d3 <= thou2) && (
eda+edb+edc+d3 <= thou3) && (et+d1+d2+d3 <= thou4))==1
 disp('watermark constraints are true ')
 else
 disp('watermark constraints are not true ')
 end
 disp('2. Check False Insertion ')
 et=ResultsC(1,1); eda=ResultsC(1,2);edb=ResultsC(1,3);
edc=ResultsC(1,4); d1=ResultsC(1,5);d2=ResultsC(1,6);d3=ResultsC(1,7);

 if ((eda+edc+d2+d3<=thou2)&&(eda+d1+d3<=thou3
)&&(eda+edb+edc+d2+d3<=thou4))==1
 disp('Attack False Insertion are true ')
 else
 disp('Attack False Insertion are not true ')
 end
 else
 disp('The watermark constraints change ')
 end

else
 MenuLKR
end
lagi=input('Do you want to repeat ? (Y/N)', 's')

```

---

---

end

---

### APPENDIX I B : GPKR Watermarking Technique Matlab Code

#### 1. *menuGPKR.m*

```

lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' MAIN MENU GENERATE WATERMARK ')
disp(' GPKR WATERMARKING TECHNIQUE ')
disp('=====')
disp('1. Set up Network GPKR watermarking technique ')
disp('2. GenerateCoverMedium ')
disp('3. Generate stream watermark')
disp('4. Computation Cover medium with out Watermark Constraints ')
disp('5. Cover medium with data message with Wateramrk constraints ')
disp('6. Extract ')
disp('7. Detecting ')
disp('8. Attack_data deletion ')
disp('9. Attack_False data insertion')
disp('10. Attack_data Modification')
disp('11. Attack_data_Replication')
disp('12. Menu ')
disp('13. Exit ')
disp('=====')
pilih = input ('Which one do you want = ');
if pilih==1
 NetworkSetupWMSN
elseif pilih==2
 GenerateCoverMedium
elseif pilih==3
 GenerateWatermark
elseif pilih==4
 ComputationCoverMedium
 elseif pilih==5
 EmbeddingWithCONSTRAINTS
elseif pilih==6
 ExtractWMSN
elseif pilih==7
 DetectingProcessWMSNs
elseif pilih==8
 DetecAttackDelete
elseif pilih==9
 DetectAttackFalseInsert
 elseif pilih==10
 DetectAttackModification
elseif pilih==11
 DetectAttackReplication
elseif pilih==12
 MenuGPKR
else
 exit
end

```

---

```

lagi=input('Anda ingin mengulangi lagi ? (Y/N)','s')
end
disp('')
disp('Copyright protection using GPKR WATERMARKING TECHNIQUE ')
disp('')

```

---

## 2. NetworkSetupWMSN.m

```

disp('=====')
disp(' GPKR WATERMARKING TECHNIQUE ')
disp(' Copyright protection of image in WMSN ')
disp(' NETWORK SET UP ')
disp('=====')
d=[];
noOfNodes = 50;
rand('state', 0);
figure(1);
clf;
hold on;
L = 200;
M = 100;
R = 30; % maximum range;
netXloc = rand(1,noOfNodes)*L;
netYloc = rand(1,noOfNodes)*M;
Coordinat1 =[];
for i = 1:noOfNodes
plot(netXloc(i), netYloc(i), 'o',...
 'MarkerEdgeColor','k',...
 'MarkerFaceColor','r',...
 'MarkerSize',10);
xlabel(' Coordinate of X axis')
ylabel(' Coordinate of Y absis ')
title('EXPERIMENT SET UP COPYRIGHT PROTECTION OF IMAGE WIRELESS
MULTIMEDIA SENSOR NETWORKs')
text(netXloc(i), netYloc(i), num2str(i));
for j =1:noOfNodes
distance = sqrt((netXloc(i) - netXloc(j))^2 + (netYloc(i) -
netYloc(j))^2);
Coordinat1(i,1)= netXloc(i)
Coordinat1(j,2)= netYloc(j)
disp(sprintf('%.4g | %.4g | %.4g ',i, netXloc(i), netYloc(j)));
if distance <= R
matrix(i, j) = 1; % there is a link;
line([netXloc(i) netXloc(j)], [netYloc(i) netYloc(j)], 'LineStyle', ':');
else
matrix(i, j) = inf;
end;
end;
end;
xlswrite('Coordinat1.xls',Coordinat1)
s1=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\COPYRIGHTProtectionofimageWMSNs\Coordinat1.xls',1,'A4:B50');
s2=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\COPYRIGHTProtectionofimageWMSNs\Coordinat1.xls',1,'A1:B3');
Coordinate2= [s1 ; s2]
xlswrite('Coordinate2.xls',Coordinate2)

```

---

---

```

s3=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\COPYRIGHTPROTECTIONOFIMAGEWMSNs\Coordinate1.xls',1,'A7:B50');
s4=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\COPYRIGHTPROTECTIONOFIMAGEWMSNs\Coordinate1.xls',1,'A1:B6');
Coordinate3= [s3 ; s4]
xlswrite('Coordinate3.xls',Coordinate3)
DATAposisi = [Coordinate1 Coordinate2 Coordinate3];
xlswrite('threeOtherNodesWMSNs.xls', DATAposisi);
disp('Generate time arrival and departure sensor ')
time=[];
for j = 1:noOfNodes
 disp('-----')
 disp('No | tDA | tDB | tDC ')
 disp('-----')
 for i=1:noOfNodes
 disp(sprintf('%.4g | %.4g | %.4g | %.4g |',i, random('unif',0,1) ,
random('unif',0,1) ,random('unif',0,1)));
 time(i,1)= random('unif',0,1);
 time(i,2)= random('unif',0,1);
 time(i,3)= random('unif',0,1);
 end
end
disp('Generate Temperature of the propagation media ')
xlswrite('time123.xls',time)
temp=[];
Max_Temp=50;
temp=rand(Max_Temp,1)*Max_Temp
xlswrite('temp.xls',temp)
disp(' Generate the feasibility of value ')
fisible=[];
for j = 1:noOfNodes
 disp('-----')
 disp('No | thou1 | thou2 | thou3 | thou4 ')
 disp('-----')
 for i=1:noOfNodes
 disp(sprintf('%.4g | %.4g | %.4g | %.4g |%.4g ',i,
random('unif',0,1) , random('unif',0,1) ,random('unif',0,1),
random('unif',0,1))) ;
 fisible(i,1)=random('unif',0,1) ;
 fisible(i,2)=random('unif',0,1) ;
 fisible(i,3)=random('unif',0,1) ;
 fisible(i,4)=random('unif',0,1) ;
 end
end
xlswrite('Fisible1234.xls',fisible)
DATAlengkap = [DATAposisi time temp fisible];
xlswrite('datalengkapWMSNs.xls',DATAlengkap);
A=xlsread ('D:\Gambar Thesis Bambang\Program Matlab untuk secure data
WSNs\COPYRIGHTPROTECTIONOFIMAGEWMSNs\datalengkapWMSNs.xls',1,'A1:Q32');
xlswrite('data32nodesWMSNs.xls',A)
lagi=input('Do you want to return ? (Y/N)', 's')
disp('')
disp('Copyright protection of the image in WMSN using GPKR watermarking
')
disp('')
end

```

---

3. *GenerateWatermark.m*

```

clc
lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' MAIN MENU GENERATE WATERMARK ')
disp(' GPKR WATERMARKING TECHNIQUE ')
disp(' 1. Applied Pyramid Transform for reducing image ')
disp(' Generating Watermark Matrix biner ')
disp(' Watermark Constrints Kolmogorov rule ')
disp(' The image reduced ')
disp(' 2. Main Menu GPKR Watermarking technique ')
disp('-----')
pilih = input ('Which one do you want = ');
if pilih==1
Original = input('Image that multimedia Sensor node capture ? ');
[g h] = size(Original);
n = input('How many times will be it reduced? ')
if n==1
 reduce1 = pyr_reduce(Original);
 imwrite(reduce1,'lastimage.jpg','jpg','Quality',10);
 [g h] = size('lastimage.jpg')
elseif n==2
 reduce2 = pyr_reduce(pyr_reduce(Original));
 imwrite(reduce2,'lastimage.jpg','jpg','Quality',10);
elseif n==3
 reduce3 = pyr_reduce(pyr_reduce(pyr_reduce(Original)));
 imwrite(reduce3,'lastimage.jpg','jpg','Quality',10);
elseif n==4
 reduce4 = pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(Original))));
 imwrite(reduce4,'lastimage.jpg','jpg','Quality',10);
elseif n==5
 reduce5 =
pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(Original))));
 imwrite(reduce5,'lastimage.jpg','jpg','Quality',10);
elseif n==6
 reduce6 =
pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(Original)
l))))));
 imwrite(reduce6,'lastimage.jpg','jpg','Quality',10);
elseif n==7
 reduce7 =
pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_red
uce(Original)))))));
 imwrite(reduce7,'lastimage.jpg','jpg','Quality',10);
elseif n==8
 reduce8 =
pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_red
uce(pyr_reduce(Original)))))));
 imwrite(reduce8,'lastimage.jpg','jpg','Quality',10);
elseif n==9
 reduce9 =
pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_red
uce(pyr_reduce(pyr_reduce(Original)))))));
 imwrite(reduce9,'lastimage.jpg','jpg','Quality',10);
elseif n==10

```



---

```

 reduce10 =
 pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_reduce(pyr_redu
 ce(pyr_reduce(pyr_reduce(pyr_reduce(Original)))))))));
 imwrite(reduce10, 'lastimage.jpg', 'jpg', 'Quality', 10);
else
end
imwrite(Original, 'OriginalImage.jpg', 'jpg', 'Quality', 10);
Last_x=imread('lastimage.jpg');
[m1 n1] = size(Last_x);
Cp = Original;
Cp(g-100:g-100+m1-1,h-100:h-100+n1-1)= Last_x(1:m1,1:n1);
%subplot(3,3,1)
%subplot(3,3,2)
subplot(3,3,3)
%imshow(Last_x(:, :, 1))
imagesc(Last_x(:, :, 1))
title('View Red picture')
subplot(3,3,4)
imshow(Original)
title('Original image ')
subplot(3,3,5)
%imshow(Last_x)
imagesc(Last_x)
title('reduced image')
subplot(3,3,6)
%imshow(Last_x(:, :, 2))
imagesc(Last_x(:, :, 2))
title('View Green picture')
%subplot(3,3,7)
%imshow(Cp)
%title('Watermarked image')
%subplot(3,3,8)
subplot(3,3,9)
%imshow(Last_x(:, :, 3))
imagesc(Last_x(:, :, 3))
title('View Blue picture')
disp('Matrix decimal from an image')
Last_x(:, :)
disp(' Convert Matrix decimal to Matrix binary ')
%Rside=de2bi(Last_x(:, :, 1), 8)
% produce Digital Message Image
%st0=reshape(Rside, 8, []);
st0=de2bi(Last_x(:, :, 1), 8);
st1=de2bi(Last_x(:, :, 2), 8);
st2=de2bi(Last_x(:, :, 3), 8);

% the key encrypting is fliplr(A) returns A with columns flipped in the
left-right direction,
% that is, about a vertical axis.
% the another key is flipud(A)
st01=fliplr(st0)
sta=fliplr(st1);
stb=flipud(st2);
disp(' Watermark Matrix Signal to be come 2 Row x Coloumn ')
st= [st01 ; sta ; stb]
%xlswrite('DMI.xls', st0); % used for extracting binary
matrix%xlswrite('WatermarkMatrixSignal.xls', st);

```

---

---

```

xlswrite('WMatrixOriginal.xls',st); % used for extracting binary
matrix%xlswrite('WatermarkMatrixSignal.xls',st);

disp(' Convert to watermark constraint by using Kolmogorov rule ')
disp('-----')
KolmogorovRuleWMSN
else
 MenuGPKR
end
lagi=input('Do you want to repeat ? (Y/N)','s')
end

```

---

#### 4. DetectingProcessWMSNs.m

```

clc
lagi='y';
while lagi=='y' | lagi=='Y';
disp('=====')
disp(' GPKR WATERMARKING TECHNIQUE ')
disp(' 1. Detecting Process Replication ')
disp(' 2. Menu GBKR Watermarking technique ')
disp('=====')
pilih=input('What Do you want = ');
if pilih==1
ResultsA=xlsread('C:\Users\14120992\Dropbox\Photos\COPYRIGHTPROTECTIONOFI
mageWMSNs\XLS\ResultsWithout.xls',1,'A1:H32');
ResultsB=xlsread('C:\Users\14120992\Dropbox\Photos\COPYRIGHTPROTECTIONOFI
mageWMSNs\XLS\ResultsWithWC.xls',1,'A1:H32');
ResultsD=xlsread('C:\Users\14120992\Dropbox\Photos\COPYRIGHTPROTECTIONOFI
mageWMSNs\XLS\ATTACKREPLICATION.xls',1,'A1:H32');

a1=sum(ResultsA,1);
X = ResultsB-ResultsA;
x1=sum(X,1);
threshold = abs((x1.*a1)/sqrt(a1.*a1));
disp('The value of threshold is = '); disp(threshold)
disp('-----')
disp('The similarity use sim(C,X) = (C*X) /sqrt(X*X) ')
c1=sum(ResultsD,1);
X2 = ResultsD-ResultsA;
x2=sum(X2,1);
similarity = abs((x2.*c1)/sqrt(c1.*c1));
disp('The value of similarity is ');
disp(similarity)

if threshold >= similarity
 disp(' Attack Replication do not change the wateramrk ')
 disp('-----')
else
 disp('The watermark constraints change ')
 disp('-----')
end
else

```

---

```
MenuGPKR
end
lagi=input('Do you want to repeat ? (Y/N)', 's')
end
```

---

---

## APPENDIX II

### List Publications

1. Harjito, Bambang and Potdar, Vidyasagar and Singh, Jaipal, 2012. Watermarking Technique for Copyright Protection of Wireless Sensor Network Data using LFSR and Kolmogorov Complexity. 10th International Conference on Advances in Mobile Computing & Multimedia, 3-5 December 2012 Bali Indonesia.
  2. Harjito, Bambang and Potdar, Vidyasagar and Singh, Jaipal, 2012. Watermarking Technique for Wireless Sensor Networks: A State of the Art technology ” SKG 2012 : The International Conference on Semantics, Knowledge and Grids Oct 22, 2012 - Oct 24, 2012, Beijing, China
  3. Harjito, Bambang and Potdar, Vidyasagar and Singh, Jaipal, 2012. Watermarking Technique for Wireless Multimedia Sensor Networks: A State of the Art” *CUBE 2012*, Proceedings of the CUBE International Information Technology Conference, September 3–5, 2012, Pages 832-840 Pune, Maharashtra, India
  4. Harjito, Bambang and Han, Song. 2010. Wireless Multimedia Sensor Networks Applications and Security Challenges, in Barolli, L. and Xhafa, F. and Uehara, M. and You, I. (ed), 5th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA 2010), Nov 4 2010, pp. 842-846. Fukuoka, Japan: IEEE.
  5. Harjito, Bambang and Han, Song and Potdar, Vidyasagar and Chang, Elizabeth and Xie, Miao. 2010. Secure Communication in Wireless Multimedia Sensor Networks using Watermarking, in Ismail, L. and Chang, E. and Karduck, A.P. (ed), IEEE international conference on digital ecosystems and technologies (DEST 2010), Apr 12 2010, pp. 640-645. Dubai, United Arab Emirates: IEEE.
-

# Watermarking Technique for Copyright Protection of Wireless Sensor Network Data using LFSR and Kolmogorov Complexity

Bambang Harjito<sup>1,2</sup>

<sup>1</sup>School of Information Systems, Curtin University, Perth, Australia

<sup>2</sup>Informatic Department, UNS Ir. Sutami No.36 A Surakarta +62271663375 Indonesia

[harjito.bambang@postgrad.curtin.edu.au](mailto:harjito.bambang@postgrad.curtin.edu.au)

Vidyasagar Potdar

School of Information Systems, Curtin Business School Curtin University Perth, Australia

[v.potdar@curtin.edu.au](mailto:v.potdar@curtin.edu.au)

Jaipal Singh

Department of Electrical & Computer Engineering Curtin University Perth, Australia

[j.singh@curtin.edu.au](mailto:j.singh@curtin.edu.au)

## ABSTRACT

The function of Wireless Sensor Networks (WSNs) is to collect and store data sent to other nodes or servers. Current technologies allow validation during transit, but stop after the data reaches its destination. One of the challenges with these technologies is to ensure that the source of the data is preserved, once it leaves the WSN. This is important as the data can be used by other applications or distributed to other parties. Therefore, it needs to be ensured that the data source is identifiable and the data is valid. Sensors are susceptible to various types of attack, such as data modification, data insertion and deletion, or even physical capture and sensor replacement. Hence, security becomes an important issue with WSNs. The traditional algorithms are used for securing data transmission between sensor nodes. However these algorithms need millions of multiplication instructions to perform operations, and cannot efficiently protect the copyright of the valuable sensor data. Watermarking is one of the effective choices to overcome this challenge. Watermark adds a second line of defense to ensure that the data is valid, even if someone cracks the encryption. This paper proposes a watermarking technique for copyright protection of data in WSNs. It also provides performance evaluation of the technique to show its robustness against various types of attacks, like data deletion, packet replication and Sybil attacks.

## Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]:

Wireless communication

## General Terms

Algorithms, Security, Human Factors

## Keywords

WSN, Digital watermarking, LFSR, Kolmogorov Rule

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) have the capabilities of sensing, processing and wireless communication, all built into a tiny embedded device [1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MoMM2012, 3-5 December, 2012, Bali, Indonesia.

Copyright 2012 ACM 978-1-4503-1307-0/12...\$15.00.

The primary function of WSNs is to collect and disseminate critical data that characterize the physical phenomena within the target area. This is primarily done by WSN nodes. WSN nodes have low power supply and limited computational capability because they operate on batteries. Given their limited power supply, it becomes challenging to ensure security. Watermarking is a lightweight security technique that has been traditionally used for providing copyright protection to multimedia data, like images and video clips. However, recent work has incorporated it in the field of WSN as well [7, 9]. There are numerous security dimensions, like authenticity, integrity, copyright protection, etc., which need to be addressed to make the WSNs secure. Watermarking techniques have been investigated to address these issues. The main reason for adopting watermarking techniques is that watermarking algorithms are shown to be less energy demanding [7, 9]. Another advantage of using watermarking is that it provides security at all times as the watermark, once embedded, becomes a relatively inseparable constituent part of the host media, unlike cryptography which provides no security once the content is decrypted [6-8]. Hence, the research in the area of watermarking and WSNs has become increasingly important. The objective of this paper is to present a watermarking technique to protect the copyright of the sensor data against possible malicious or accidental tampering of data.

## 2. BACKGROUND OF WATERMARKING

In this section, we briefly explain how watermarking techniques operate. Watermarking technique is the process of embedding information which allows an individual to add hidden copyright notices or other verification messages to digital audio, video, or image signals and document objects [2-4]. Watermarking technique as a communication task consists of three main stages: watermark generation process, watermark embedding process which includes information transmission and possible attacks through the communication channel, and watermark detection and retrieval process. Watermark generation process is critical as its requirements are unique and complex. The watermark message must contain information that is unique, such as simple text [5] [6]. The key embedding must also be unique in order to make a secrecy key, such as a binary stream [7] [8] [9] [10] [11]. Both the watermark message and the key embedding are used as the input and processed in the watermark generator to produce a

watermark signal. The watermark embedding process is undertaken by an embedder. The embedder combines the cover medium, the watermark signal, and the sensed data and key embedding, to create a watermarked cover medium. During transmission, there are many things that interfere with the communication process, such as noise, that decreases the quality of transmission and leads to the watermarked cover medium being dropped. The other things are watermark attacks, such as cropping, compression, and filtering. The aim of these attacks is to remove the watermark signal from the watermarked cover medium. The extraction and detection process is undertaken by a detector. The detection process consists of an extraction unit to first extract the watermark signal, and later compare it with the cover medium without the watermark signal. The extraction process can be divided into two phases - locating the watermark and recovering the watermark information.

### 3. RELATED WORK

Over the last decade, a lot of research has been carried out on digital watermarking techniques for texts, images, audios, videos and even relational databases [12-15]. Our research group itself has some of the earliest works to its credit in the areas of watermarking for RFID [34-48]. But there have been relatively few works on digital watermarking techniques for WNSs [5] [16, 17]. Feng et al. [5] developed the first system of watermarking technique to embed cryptologically encoded authorship signatures into the data and information acquired by wireless embedded sensor networks for copyright digital ownership. Later, Sion et al. [16] provided a way for copyright protection to data stream owners and the authorized users. For example, consider a case where a stream is generated and safely transmitted from the sensors to the base station. A watermark is applied to the stream at the base station. The data are then transmitted to an authorized user. Now, in case of an attack, the owner and the authorized users need a way to prove that the data were generated by them and the stream was illegally obtained by the attacker. One commonly accepted way to prove ownership is using embedded watermarks. This technique works by embedding a watermark bit into major extremes, the extremes that survive any uniform sampling. Koushanfar et al. [6] present an active watermarking technique that can be used on the data to be processed during the common sensor fusion application, using sensors of different modalities. This technique has been used for copyright protection. Xiao et al. [10] have proposed a watermarking technique for protecting copyright by taking advantage of the characteristic of the sending time. Based on digital watermarking, Zhang et al. [18] have put forward an end-to-end, watermark statistical approach for data authentication that provides inherent support for in-network processing. In this technique, the authentication information is modulated as watermark, and embedded into the sensory data at the sensor nodes. A communication protocol for WSNs has been introduced by Xuejun et al. [19] to authenticate sensitive data transmission. The technique uses sensitive information as watermark. The watermark is then embedded into sensory data in the sensor nodes. A threshold is used to avoid the alteration of the lowest bit is "1", append "1" into the Output Binary system (OBS), otherwise, append "0" to the OBS and make a big influence on the precision of the sensory data. Kamel et al. [20] introduce a technique to provide for data integrity. This technique, based on distortion free watermarking, embeds the watermark in the order of the data element, so that it does not cause distortion of the data. Albath et al. [8] have introduced a method to provide protection against passive eavesdropping by

employing confidential transmission of data messages. Table 1 shows a summary of the watermarking techniques that can be used in WSNs. Although some research works have attempted to apply digital watermarking technique to wireless sensor networks for security, authentication and integrity purposes, most of the existing studies were conducted for copyright protection purposes.

**Table 1 Watermark embedding approaches and their purpose**

| Author                  | Watermark embedding technique                                                                                                                               | Purpose              |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| Feng et al. [5]         | Adding watermark constraint to the processing step during the network operation                                                                             | Copyright            |
| Sion et al. [16]        | Selection criteria using MSB                                                                                                                                | Copyright            |
| Koushanfar et al. [6]   | Adding watermark constraint to the processing step during the network operation                                                                             | Copyright            |
| Xiao et al. [10]        | Modification of the embedding bit of each packet.                                                                                                           | Copyright            |
| Zhang et al. [18]       | The watermark sensory data, $d(x,y) = w(x,y)+o(x,y)$ , $w(x,y)$ is the watermark for sensor node and $O(x,y)$ is the sensory data                           | Authentication       |
| Xuejun et al. [19]      | IIS = input integer stream, IBS=input binary stream. T = Threshold, If IIS $\geq$ T "IBS=1" become "IBS=0" Else "IBS=0" become "IBS=0"                      | Authentication       |
| Kamel et al. [7]        | Concatenation of the current group hash value group $g_i$ and next group hash value group $g_{i+1}$ . $W_i = HASH(K    g_i    SN)$<br>$SN = serial\ number$ | Integrity            |
| Julia Albath et al. [8] | generating the one-time pad by repeatedly concatenating the substring                                                                                       | Secure Communication |

Here we present a watermarking technique for copyright data protection based on Linear Feedback Shift Register (LFSR) and Kolmogorov rule. Details of the technique are discussed in Section 4.

### 4. PROBLEM DESCRIPTION

In the application of wireless sensor networks, all the communication between different nodes occurs in the broadcast fashion through the transmission channel, where any node may become an attack target with external and internal security risks, including eavesdropping, leak, data tampering, etc. In the particular application fields, if the data transmission is not reliable, the security of the whole network is affected. In order to make the network secure against the attackers, secure data transmission between sensor nodes is used. Secure data sensor networks use many cryptographic algorithms. These techniques need a large number of multiplication instructions in order to perform operations [21-25]. However, they cannot efficiently protect the copyright of the valuable sensor data. To address this issue, copyright protection for the valuable sensory data has become an important issue because a malicious adversary can easily duplicate the segments of the valuable sensory data for taking advantage [26]. In the previous section, we conducted an in-depth literature survey of the watermarking approaches in WSNs and their purposes. As identified by us, many researches

have applied the digital watermarking technique to WSNs for security purposes [8], authentication purposes [18], [19] and integrity purposes [7]. Many other researches [5] [6] [10] [16] have worked on copyright protection. However, there has been no work that uses LFSR and Kolmogorov rule for copyright protection. This paper will show that the use of the LFSR and Kolmogorov technique provides a better protection of watermarks, compared to other watermarking techniques.

## 5. PROPOSED WATERMARKING TECHNIQUE

In this section, we give a general overview of our proposed watermarking technique to ensure the reliability of copyright data protection. The copyright data protection model based on watermarking technique can be seen in Figure 1. This model consists of four steps: (1) Cover medium generation process (2) Watermark generation process (3) Embedding process, and (4) Detection and extraction process. The cover medium generation process generates a cover medium by using an atomic trilateration process. The watermark generation process creates watermark constraints and message sensed data. This process requires a sensed data through the LFSR process, partitioned process and Kolmogorov rule process. The embedding process generates a watermarked cover medium and the detection process detects the watermark signal.

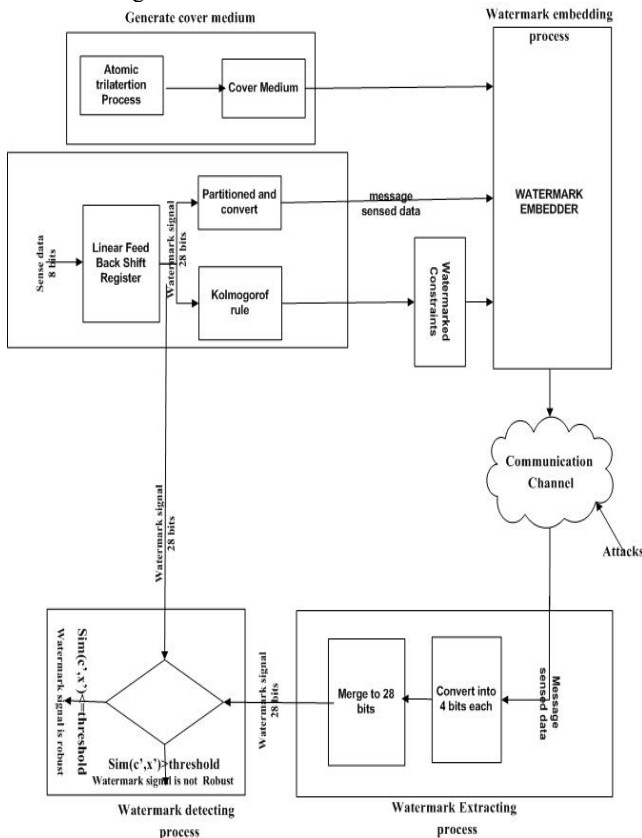


Figure 1. Copyright data protection based on watermarking

### 5.1 Cover medium generation process

In this section, we explain the process of generating cover medium by using the atomic trilateration process (Pseudo code 1). With respect to two-dimensional sensor networks, atomic trilateration is a well-known procedure by which a sensor node in

a network can determine its position by using the distance and position of three other sensor nodes of known location. From these distances and positions, a sensor node which is trying to determine its location can generate a non-linear system programming.

#### Pseudocode 1. Generate Cover Medium

**Input:**  $(x_A, y_A), (x_B, y_B), (x_C, y_C), T_c, t_{DA}, t_{DB}, t_{DC}, (x_D, y_D), \varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \delta_3$

**Output:** The cover medium is

$$\min f = \varepsilon_t + \varepsilon_{DA} + \varepsilon_{DB} + \varepsilon_{DC} + \delta_1 + \delta_2 + \delta_3$$

Constraints:

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(t_{DA} + \varepsilon_{DA}) \leq \delta_1$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(t_{DB} + \varepsilon_{DB}) \leq \delta_2$$

$$\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(t_{DC} + \varepsilon_{DC}) \leq \delta_3$$

#### Steps:

1. Compute  $V_s = 331.4 + 0.6T_c$
2. Compute  $d_{DA} = V_s * t_{DA}$ ,  $d_{DB} = V_s * t_{DB}$ ,  $d_{DC} = V_s * t_{DC}$ .  
Where  $d_{DA}$ ,  $d_{DB}$  and  $d_{DC}$  are between nodes  $D$  and the sensor nodes are then measured using TDoA.
3. Append  $\varepsilon_t$  error of measurement time to step (2)
4. Append  $\varepsilon_{DA}$ ,  $\varepsilon_{DB}$ , and  $\varepsilon_{DC}$  errors of measurement distance to step (2).
5. Compute
$$d_{DA} = \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}$$

$$d_{DB} = \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2}$$

$$d_{DC} = \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}$$
6. Append  $\delta_1, \delta_2$  and  $\delta_3$  errors between the Euclidean distances step (3)
7. Replace  $d_{DA}$ ,  $d_{DB}$  and  $d_{DC}$  from step (2) to step (3) and then compute them.
8. Print cover medium

### 5.2 Watermark generation process

Watermark generation process involves four main stages - sensitive data conversion, watermark signal generation, watermark constraint generation, and message sensed data generation.

#### 5.2.1. Converting sensitive data into binary sequence

The first step is converting sensitive data into binary sequence. Any data which compromises confidentiality, integrity, and/or availability could have a materially adverse effect on. Such data is called sensitive data. The sensitive data is directly proportional to the materiality of the compromised data with respect to these criteria. Sensitive data can be formed in the form of video, audio, image and scalar data. Shih et al. [27] present their findings on sensitive data and privacy issues of applications in Body Sensor Networks (BSNs). In BSNs, the applications collect sensitive physiological data of the user and send them to other parties for further analysis. The sensitive data are ECG signals, and blood

pressure. These data are required to be protected and converted from scalar data into binary stream.

### 5.2.2 Generating watermark signal using Linear Feedback Shift Register (LFSR)

One method of forming a binary sequence for generating watermark is to apply LFSR whose characteristic polynomial is primitive [28, 29]. LFSR is a shift register whose input bit is a linear function of its previous state. The only linear function of single bits is exclusive-or (*xor*), therefore it is a shift register whose input bit is driven by *xor* of some bits of the overall shift register value.

LFSR can be defined by a recurrence relation:

$$s_{K+n} = \sum_{i=0}^{n-1} c_i s_{k+1}, \text{ where } k \geq 0, n \in Z \text{ and } c_i \text{ are binary constants such that } c_0 = 1.$$

associated with such a recurrence relation is a binary polynomial

$$f(x) = c_0 + c_1x + \dots + c_{k-1}x^{k-1} + x^k,$$

called the characteristic polynomial of the LFSR. The coefficient  $c_i$  is feedback constant. Such sequence can be mechanized by using an LFSR whose tap settings are defined by the feedback constant.

#### Pseudo code 2. Generate watermark signal

**Input:** Sensed data, coefficients  $c_i$  of the binary polynomial as watermark key

**Output:** 28 bits watermark signal

#### Steps:

1. Convert sensed data into binary sequence.
2. Use the coefficients  $c_i$  of the binary polynomial  $f(x)$  as watermark key
3. Generate an infinite binary sequence using the coefficient  $c_i$  into a LFSR ( $s_{K+n}$ ).
4. Cut the infinite binary sequence from 1 to 28 as watermark signal.
5. Print 28 bits watermark signal

### 5.2.3 Kolmogorov rule to create watermark constraints

Andrew Nikolaevich Kolmogorov [30] states that the complexity of an object is the length of the shortest computer program that can reproduce the object. The Kolmogorov complexity is defined as a probability distribution under which the worst-case and average-case running time are the same. We know that the Kolmogorov complexity rule is the short description length of the overall description interpreted by the computer. The three papers [5, 31, 32] use the Kolmogorov rule for numbering the variables of the linear combination in the optimization objective function, and a set of constraints. We have also used the Kolmogorov rule which can be seen in Table 1.

Table 1 Kolmogorov rule

|              |                 |                 |                 |            |            |            |
|--------------|-----------------|-----------------|-----------------|------------|------------|------------|
| 1            | 2               | 3               | 4               | 5          | 6          | 7          |
| $\epsilon_t$ | $\epsilon_{DA}$ | $\epsilon_{DB}$ | $\epsilon_{DC}$ | $\delta_1$ | $\delta_2$ | $\delta_3$ |

#### Pseudo code 3. Generate watermark constraints

**Input:** 28 bits watermark signal

**Output:** Watermark constraints

#### Steps:

1. Group 28 bit watermark signals into groups of 7 bits each.

2. Match the bit number with corresponding variable number from table 2.
3. If a bit is assigned a variable within a group, that variable is included in the linear.
4. Else a bit zero is assigned to the variable within the group and that variable is not included in the linear.
5. Go to 2.
6. Print watermark constraints.

### 5.2.4 Partitioning and conversion to create message sensed data

In this section, we explain how a message sensed data can be created. To create this message sensed data, 28 bit watermark binary, resulting from the generate watermark signal, has been used.

#### Pseudo code 4. Generate create message data

**Input:** 28 bits watermark signal

**Output:** message sensed data

#### Steps:

1. Group 28 bit watermark signals into groups of 4 bits each.
2. Convert each group into decimal number to get weight factors.
3. Print message sensed data

## 5.3 Watermark Embedding Process

Embedding process is the second step of the watermarking system undertaken by a watermark embedder. The embedder combines the cover medium, the watermark constraints and the message sensed data, to create a watermarked cover medium. The watermarked cover medium is perceptually identical to the cover medium. The process of watermark embedding can be shown in Figure 1.

#### Pseudo code 5. The process of embedding

**Input:** cover medium, 28 bits watermark signal, Watermark constraints, message sensed data.

**Output:**  $(x_D, y_D)$ ,  $\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$ ,  $\delta_1, \delta_2, \delta_3$  and min f

#### Steps:

1. Generate  $(x_A, y_A), (x_B, y_B)$  and  $(x_C, y_C)$  using uniform distribution on interval  $[0, 1]$ .
2. Generate  $\epsilon_{DA}, \epsilon_{DB}$ , and  $\epsilon_{DC}$  using uniform distribution on interval  $[0.02, 0.1]$ .
3. Generate  $\delta_1, \delta_2$  and  $\delta_3$  using gauss distribution on interval  $[0, 1]$ .
4. Generate  $\tau_1, \tau_2, \tau_3$  and  $\tau_4$  using gauss distribution on interval  $[0, 1]$ , so that these value do not harm the feasibility of the solution of the cover medium
5. Generate  $T_C$  using gauss distribution on interval  $[0, 1]$ .
6. Change coefficient objective f to weight factor of message sensed data respectively.
7. Append watermark constraints into cover medium
8. Compute and print  $(x_D, y_D)$ ,  $\epsilon_t, \epsilon_{DA}, \epsilon_{DB}, \epsilon_{DC}$ ,  $\delta_1, \delta_2, \delta_3$  and min f

## 5.4 Watermark Extraction

The extraction process is also undertaken in the watermark detector as we want to recover the message sensed data from the cover medium. The process of watermark extracting can be shown in Figure 1.



**Pseudo code 6. The process of extracting watermark signal**

**Input :**  $\varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \delta_3$

**Output :** 28 bits watermark signal

**Steps :**

1. Compute the value of the objective f using  $\varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \text{ and } \delta_3$
2. If the value of the objective does not change go to 3
3. Else go to step 1.
4. Take the coefficients of objective f.
5. Convert the coefficient of objective f into 4 bits each.
6. Merge all of these 4 bits to 28 bits
7. Print 28 bits watermark signal.

**5.5 Watermark Detection**

The end of the watermarking system is the detection process which is a crucial part which allows the sender to identify and provide information to the intended receiver. The detection process is undertaken by a detector. There are two types of detection: informed detection and blind detection according to whether the cover medium is needed or not in the detection process. The process of detecting watermark has not been explained in Feng et al. [5] or Koushanfar et al. [6]. Both of them are explain the process of embedding the watermark. To verify the presence of the watermark, we adopt the concept of Cox *et al* [33]. They draw parallels between their technology and the spread-spectrum communication, since the watermark is spread over a set of visually important frequency components. Let  $x$  be the error of the optimal solution with message sensed data,  $x'$  be the error of the optimal solution with the message sensed data and watermark constraints, and  $x''$  be the error of the optimal solution with the message sensed data and watermark constraints attack. For detecting the watermark, a correlation value or similarity measure is used in most of these methods. Here, to verify the presence of the watermark signal, the similarity measure between the normalized difference errors of the optimal solution with message sensed data and watermark constraints, and the optimal solution with message sensed data is ( $c = x' - x$ ). The similarity measure between the normalized difference errors of the optimal solution with message sensed data and watermark constraints attacks and the optimal solution with message sensed data is ( $c' = x'' - x$ ). The similarity measure is given by the normalized correlation

$$\text{coefficient } sim(C', X') = \frac{C' \cdot X'}{\sqrt{X' \cdot X'}}$$

**Pseudo code 7. The process of detecting watermark signal**

**Input:** Watermark signal by LFSR and watermark signal by process of extracting  $x = [\varepsilon_t, \varepsilon_{DA}, \varepsilon_{DB}, \varepsilon_{DC}, \delta_1, \delta_2, \delta_3]$ ,  $x' = [\varepsilon_t', \varepsilon_{DA}', \varepsilon_{DB}', \varepsilon_{DC}', \delta_1', \delta_2', \delta_3']$ , and  $x'' = [\varepsilon_t'', \varepsilon_{DA}'', \varepsilon_{DB}'', \varepsilon_{DC}'', \delta_1'', \delta_2'', \delta_3'']$

**Output:** Watermark signal by LFSR robust or not robust

**Steps:**

1. Compute  $c = x' - x$
2. Compute  $c' = x'' - x$

3. Compute normalized correlation the results of error the cover medium with watermark constraints  $threshold = \frac{C \cdot X'}{\sqrt{X' \cdot X'}}$

4. Compute normalized correlation the results of error the cover medium with watermark constraints attacks  $sim(C', X') = \frac{C' \cdot X'}{\sqrt{X' \cdot X'}}$ .

5. If  $threshold > sim(C', X')$  go to 7
6. Else  $threshold < sim(C', X')$  go to 8 watermark signal by LFSR is not robust
7. Print watermark signal by LFSR is robust
8. Print watermark signal by LFSR is not robust

**6. EXPERIMENT SETUP**

This section describes the experiment setup used for an extensive testing for the purpose of the copyright data protection model, based on watermarking technique. In this experiment setup we used TOMLAB. It is a general purpose development environment in MATLAB for research and practical solution of optimization problems. It has grown out of the need for advanced, robust and reliable tools to be used in the development of algorithms and software for the solution of many different types of applied optimization problems.

**6.1 Network Setup**

In this section we will use the scenario of the atomic trilateration process as shown in Figure 2.

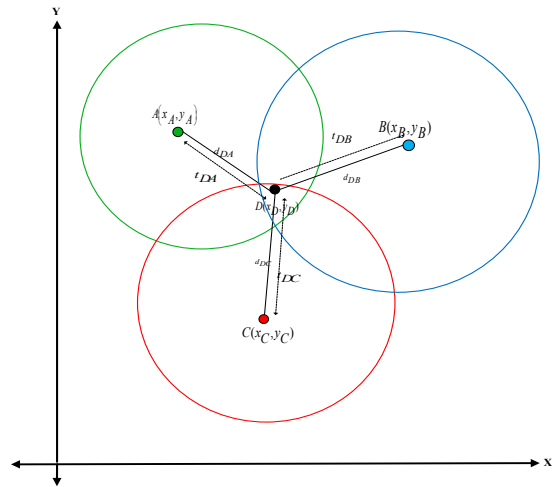


Figure 2. Atomic trilateration

Sensor Node D trilaterates with another three sensor nodes A, B, and C whose coordinates are  $(x_A, y_A)$ ,  $(x_B, y_B)$ , and  $(x_C, y_C)$ . The distance is computed using time differences of arrival (TDoA) between acoustic signals simultaneously, which are emitted from a sensor nodes and received at the node D and radio frequency (RF). The sensor node D turns on a timer upon receiving the RF signal from the sensor node to measure the difference between the arrival of the RF and acoustic signals from that sensor node. The time measurements have an error. The speed of the acoustic signal is a function of the temperature of the propagation media. The relationship between the speed of the

acoustic signal  $V_s$  (m/s) and the temperature  $T_c$  is as follows:

$$V_s = 331.4 + 0.6T_c \quad (1)$$

By using the pseudo code 1, we find that the objective function is to minimize the overall error in the system, and can be stated as shown in Equation (2):

Objective Function

$$\min f = \varepsilon_t + \varepsilon_{DA} + \varepsilon_{DB} + \varepsilon_{DC} + \delta_1 + \delta_2 + \delta_3$$

Constraints :

$$\begin{aligned} & \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DA} + \varepsilon_{DA}) \leq \delta_1 \\ & \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DB} + \varepsilon_{DB}) \leq \delta_2 \quad (2) \\ & \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DC} + \varepsilon_{DC}) \leq \delta_3 \end{aligned}$$

## 6.2 Performance Metrics

The existing performance of the watermarking technique for copyright data protection is evaluated against the following performance metrics:

| Parameter                                                        | explain                                                                                                                                      | Metric     | Value                                                                                              |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------|
| Node Sensor                                                      | Number of sensor node                                                                                                                        | Integer    | 100                                                                                                |
| $(x_i, y_j)$<br>$i = j = 1, 2, \dots, n$                         | Position of two-dimensional sensor networks                                                                                                  | Coordinate | $x_i = 0.00824227,$<br>$y_i = 0.155247$                                                            |
| $T_c$                                                            | the temperature of the propagation media                                                                                                     | Degree     | 0.0205345                                                                                          |
| $t_{DA}, t_{DB}$<br>$t_{DC}$                                     | time transmission between node D to A, D to B and D to C                                                                                     | second     | $t_{DA} = 0.0220793$<br>$t_{DB} = 0.086288$<br>$t_{DC} = 0.0717121$                                |
| $V_s$                                                            | Speed acoustic signal                                                                                                                        | (m/s)      | $V_s \geq 331.4$                                                                                   |
| $\varepsilon_t$                                                  | the error in the measurement of the temperature                                                                                              | -          | $\varepsilon_t = 0,$                                                                               |
| $\varepsilon_{DA},$<br>$\varepsilon_{DB},$<br>$\varepsilon_{DC}$ | the error in the measurement of the timer from D to A, D to B and D to C                                                                     | -          | $\varepsilon_{DA} = 0.0473$<br>$\varepsilon_{DB} = -0.0141,$<br>$\varepsilon_{DC} = 0$             |
| $\delta_1, \delta_2,$<br>$\delta_3$                              | the error in the measurement between the Euclidean measurement and the measured using time differences of optimal D to A, D to B and D to C. | -          | $\delta_1 = 0, \delta_2 = 0$<br>$\delta_3 = 0$                                                     |
| $\tau_1, \tau_2, \tau_3$<br>$\tau_4$                             | the values are selected such that the feasibility of the solution space of the optimization problem is not harmed                            | -          | $\tau_1 = 0.95184611$<br>$\tau_2 = 0.16947616,$<br>$\tau_3 = 0.24915965$<br>$\tau_4 = 0.992920660$ |
| Sensed data                                                      | Data sensed by a sensor node                                                                                                                 | Bit        | 01111000                                                                                           |

| Watermark signal    | Result from LFSR                                                                               | Bit     | 1101000100001101010010100011 |
|---------------------|------------------------------------------------------------------------------------------------|---------|------------------------------|
| Message sensed data | Result from pseudo code 4                                                                      | Integer | [13 1 0 11 2 10 4 ]          |
| <i>threshold</i>    | normalized correlation the results of error the cover medium with watermark constraints        | -       | 0.001318515699954            |
| <i>sim(C', X')</i>  | normalized correlation the results of error the cover medium with watermark constraints attack | -       | 0.001084890606725            |

## 7. EXPERIMENT AND RESULTS

In this section we discuss our experiment and the results obtained. We assume that the watermark constraints estimated by the attacker are modified and changed. The corresponding watermark constraint attacks that we perform in our experiment are:

(1) deletion of a number of watermark constraints in a hope to find new results of errors of the cover medium, (2) replication of different watermark constraints generated by the LFSR, hoping to find the new results of the errors of the cover medium that will be mapped into the existing solution, (3) presenting more than one attacker identity within the network by creating more than one watermark constraint, hoping to find further results of the errors of the cover medium.

### 7.1 Data Deletion Attack

Data deletion attack is similar to the spoofed data attack in the sense that deleting watermark constraints make the error results of the cover medium invalid. The watermark signal will also be invalid because it will not approximate to the results of the errors of the cover medium without attack. If the attacker deletes watermark constraints, the receiver will not get appropriate results of errors. The data deletion attack can take place by dropping individual watermark constraint readings, or one or more watermark constraints, and preventing them from reaching the intended recipient.

Objective function:

$$\min f = 13\varepsilon_t + 1\varepsilon_{DA} + 0\varepsilon_{DB} + 1\varepsilon_{DC} + 4\delta_1 + 10\delta_2 + 3\delta_3$$

Constraints:

$$|\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DA} + \varepsilon_{DA})| \leq \delta_1$$

$$|\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DB} + \varepsilon_{DB})| \leq \delta_2$$

$$|\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t))(\Delta t_{DC} + \varepsilon_{DC})| \leq \delta_3$$

$$\begin{aligned} & \varepsilon_t + \varepsilon_{DA} + \varepsilon_{DC} \leq \tau_1 \\ & \varepsilon_t + \delta_2 \leq \tau_2 \\ & \varepsilon_t + \varepsilon_{DA} + \delta_1 + \delta_3 \leq \tau_3 \\ & \varepsilon_{DA} + \delta_1 \leq \tau_4 \end{aligned} \quad \left. \begin{array}{l} \text{Watermark} \\ \text{Constraints} \end{array} \right\} \begin{array}{l} \text{Delete} \\ \text{Constraints} \end{array}$$

$$\begin{aligned} & \varepsilon_t + 2\delta_2 \leq \tau_1 \\ & \varepsilon_{DC} + 2\delta_2 + \delta_3 \leq \tau_2 \\ & \varepsilon_{DB} + 3\delta_1 + \delta_3 \leq \tau_3 \\ & \varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4 \end{aligned} \quad \left. \begin{array}{l} \text{Redundant} \\ \text{Constraints} \end{array} \right\}$$

We get the results of the error of the cover medium by deleting watermark constraints:  $\varepsilon_t = 0,$   $\varepsilon_{DA} = 0.047454140392026,$

$$\varepsilon_{DB} = -0.013888456874134, \quad \varepsilon_{DC} = 0.000000000000137,$$

$$\delta_1 = \delta_2 = 0 \text{ and } \delta_3 = 0.013888456874134.$$

Implementing a pseudo-code 7, we conclude that the value of similarity is greater than the value of threshold: the value of similarity = 0.001084890606725 < the value of threshold = 0.001318515699954. This means that the watermark signal is robust enough to delete the attack.

## 7.2 Replication Attack

Conceptually, data replication attack is quite simple: an attacker seeks to add new constraints to the cover medium by replicating the new constraints with the existing constraints. New constraints replicated in this fashion can severely disrupt this solution of the cover medium's performance: the new results of errors of the cover medium cannot be approximated to the results of errors of the cover medium before attack. By inserting the replicated constraints along with the new constraints into the existing constraints, the attacker could easily manipulate a specific segment of the network, perhaps by disconnecting it altogether.

Objective function:

$$\min f = 13\varepsilon_t + 1\varepsilon_{DA} + 0\varepsilon_{DB} + 11\varepsilon_{DC} + 4\delta_1 + 10\delta_2 + 3\delta_3$$

Constraints:

$$|\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DA} + \varepsilon_{DA}))| \leq \delta_1$$

$$|\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DB} + \varepsilon_{DB}))| \leq \delta_2$$

$$|\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DC} + \varepsilon_{DC}))| \leq \delta_3$$

$$\left. \begin{array}{l} \varepsilon_t + \varepsilon_{DA} + \varepsilon_{DC} \leq \tau_1 \\ \varepsilon_t + \delta_2 \leq \tau_2 \\ \varepsilon_t + \varepsilon_{DA} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DA} + \delta_1 \leq \tau_4 \end{array} \right\} \text{Watermark Constraints}$$

$$\left. \begin{array}{l} \varepsilon_t + 2\delta_2 \leq \tau_1 \\ \varepsilon_{DC} + 2\delta_2 + \delta_3 \leq \tau_2 \\ \varepsilon_{DB} + 3\delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4 \end{array} \right\} \text{Redundant Constraints}$$

$$\left. \begin{array}{l} \varepsilon_{DC} + \delta_1 + \delta_3 \leq \tau_1 \\ \varepsilon_{DC} + \delta_1 + \delta_3 \leq \tau_1 \\ \varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_2 \\ \varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_2 \\ \varepsilon_t + \varepsilon_{DB} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_t + \varepsilon_{DB} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_t + \varepsilon_{DC} + \delta_3 \leq \tau_4 \\ \varepsilon_t + \varepsilon_{DC} + \delta_3 \leq \tau_4 \end{array} \right\} \text{Replication Constraints}$$

We get the results of errors of the cover medium with data replication of watermark constraints:  $\varepsilon_t = 0$ ,  $\varepsilon_{DA} = 0.047805939221795$ ,  $\varepsilon_{DB} = -0.013734653005476$ ,  $\varepsilon_{DC} = 0.000000000000137$ ,  $\delta_1 = \delta_2 = 0$  and  $\delta_3 = 0.013734653005476$ .

Implementing a pseudo-code 7, we conclude that the value of similarity is greater than the value of the threshold: the value of similarity = 0.001318515699953 < the value of threshold =

0.001318515699954. This means that the watermark signal is robust enough to thwart replication attack.

## 7.3 Sybil Attack

A Sybil attack data occurs when the attacker creates multiple identities and exploits them in order to manipulate a reputation score. The Sybil attack data is defined as a malicious device illegitimately taking on multiple data identities. The Sybil attack data in communication channel watermarking is an attack wherein a reputation network system is subverted by forging more than one identity constraints in the cover medium. Thus, in a Sybil attack, an attacker subverts the reputation network system by creating more than one constraint, and uses them to gain a disproportionately large influence. A reputation network system's vulnerability to a Sybil attack depends on how the constraint identities can be generated, the degree to which the reputation network system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation network system treats all entities identically. The objective of the Sybil data attack is to find the results of errors of the cover medium.

Objective function:

$$\min f = 13\varepsilon_t + 1\varepsilon_{DA} + 0\varepsilon_{DB} + 11\varepsilon_{DC} + 4\delta_1 + 10\delta_2 + 3\delta_3$$

Constraints:

$$|\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DA} + \varepsilon_{DA}))| \leq \delta_1$$

$$|\sqrt{(x_D - x_B)^2 + (y_D - y_B)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DB} + \varepsilon_{DB}))| \leq \delta_2$$

$$|\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2} - (331.4 + 0.6(T_c + \varepsilon_t)(\Delta t_{DC} + \varepsilon_{DC}))| \leq \delta_3$$

$$\left. \begin{array}{l} \varepsilon_t + \varepsilon_{DA} + \varepsilon_{DC} \leq \tau_1 \\ \varepsilon_t + \delta_2 \leq \tau_2 \\ \varepsilon_t + \varepsilon_{DA} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DA} + \delta_1 \leq \tau_4 \end{array} \right\} \text{Watermark Constraints}$$

$$\left. \begin{array}{l} \varepsilon_t + 2\delta_2 \leq \tau_1 \\ \varepsilon_{DC} + 2\delta_2 + \delta_3 \leq \tau_2 \\ \varepsilon_{DB} + 3\delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DB} + \delta_2 + \delta_3 \leq \tau_4 \end{array} \right\} \text{Redundant Constraints}$$

$$\left. \begin{array}{l} \varepsilon_{DC} + \delta_1 + \delta_3 \leq \tau_2 \\ \varepsilon_{DB} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DB} + 2\delta_2 + \delta_3 \leq \tau_4 \\ \varepsilon_{DC} + \delta_1 + \delta_3 \leq \tau_2 \\ \varepsilon_{DB} + \delta_1 + \delta_3 \leq \tau_3 \\ \varepsilon_{DB} + 2\delta_2 + \delta_3 \leq \tau_4 \end{array} \right\} \text{Two Identity attacker}$$

We get the results of errors of the cover medium by inserting false watermark constraints which are:  $\varepsilon_t = 0$ ,  $\varepsilon_{DA} = 0.047764846656582$ ,  $\varepsilon_{DB} = -0.013776016543383$ ,  $\varepsilon_{DC} = 0.000000000000064$ ,  $\delta_1 = 0$ ,  $\delta_2 = 0.000000000000001$ , and

$\delta_3 = 0.013776016543322$ . Implementing a pseudo-code 7, we conclude that the value of similarity is greater than the value of threshold: the value of similarity = 0.001295020739905 < the value of threshold = 0.001318515699954. This means that the watermark signal is robust enough to foil Sybil attack.

The results of these experiments have been shown in Table 2.

**Table 2. The robustness of the watermark constraints, and watermark signals**

| No | Kind of attacks     | Watermark constraints | Watermark Signal |
|----|---------------------|-----------------------|------------------|
| 1. | Data deletion       | Not change            | Robust           |
| 2. | Packet replication. | Not change            | Robust           |
| 3. | Sybil attack        | Not change            | Robust           |

## 8. PERFORMANCE EVALUATION

Next, we perform a comparative analysis of our technique with other techniques proposed by different researchers. The results of this comparative analysis are given in Table 3.

**Table 3. A comparative analysis with other approaches**

| Kind of attacks     | Zhang et al. [18] | Xuejun et al. [19] | Kamel et al. [7] | Harijito |
|---------------------|-------------------|--------------------|------------------|----------|
| Data deletion       | x                 | x                  | √                | √        |
| Packet replication. | x                 | x                  | x                | √        |
| Sybil attack        | x                 | x                  | x                | √        |

√ provides copyright data protection

x does not provide copyright data protection

In this analysis, we compared 4 approaches in terms of data deletion, packet replication, and Sybil attack. Zhang et al. approach [18] and Xuejun et al. [19] do not provide data deletion, packet replication, and Sybil attack. Kamel et al. [7] approach provides data deletion. However it does not provide packet replication and Sybil attack. Our approach provides copyright data protection against data deletion, packet replication and Sybil attacks.

## 9. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a watermarking technique for copyright data protection in WSNs. Our strategy aims at protecting copyright of data transmission between sensor nodes in WSNs against a variety of attacks, such as data modification, data deletion, packet replication, Sybil attack, false data insertion, and selective forwarding. We have not discussed some types of attacks, such as physical attack, node malfunction and Denial of service attack. We have verified that our technique can protect copyright data against deletion, packet replication and Sybil attacks. However, it cannot protect copyright data against false data insertion, data modification and selective forwarding. Therefore, we still need to improve our technique considering various circumstances in which attackers launch different kinds of attacks, for the future work.

## 10. REFERENCES

- [1] Yick, J., B. Mukherjee, and D. Ghosal, Wireless sensor network survey. *Computer Networks*, 2008. 52(12): p. 2292-2330.
- [2] Wang, X.-Y., Z.-H. Xu, and H.-Y. Yang, A robust image watermarking algorithm using SVR detection. *Expert Systems with Applications*, 2009. 36(5): p. 9056-9064.
- [3] Potdar, V., Subjective and Objective Watermark Detection Using a Novel Approach—Barcode Watermarking ed. C.I.A. Security. 2007. 576.
- [4] Potdar V., Jones, C., Chang, E, Multiple image watermarking using the SILE approach, in Proceedings of the 6th WSEAS international conference on Multimedia systems; signal processing. 2006, World Scientific and Engineering Academy and Society (WSEAS): Hangzhou, China.
- [5] Fang Jessica, P., Miodrag Real-time watermarking techniques for sensor networks Proceedings-SPIE The International Society for optical Engineering 2003(ISSU 5020): p. 391-402
- [6] F. Koushanfar, M.P., Watermarking Technique for Sensor Networks: Foundations and Applications. Book chapter, in 'Security in Sensor Networks', Yang Xiao 2007.
- [7] Kamel, I., A Lightweight Data Integrity Scheme for Sensor Networks. *Sensors*, 2011. 11(4): p. 4118.
- [8] Albath, J., Practical algorithm for data security (PADS) in wireless sensor networks Proceedings of the 6th CM international workshop on Data engineering for wireless and mobile access - MobiDE '07. 2007. 9.
- [9] Juma, H.K., I.Kaya, L. Watermarking sensor data for protecting the integrity. in Innovations in Information Technology, 2008. IIT 2008. International Conference on. 2008.
- [10] Rong, X., S. Xingming, and Y. Ying. Copyright Protection in Wireless Sensor Networks by Watermarking. in Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHMSP '08 International Conference on. 2008.
- [11] Xiaomei, D.X., Li. An Authentication Method for Sensor Nodes Based on Watermarking in Wireless Sensor Networks. in Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on. 2009.
- [12] Jian, L. and H. Xiangjian. A Review Study on Digital Watermarking. in Information and Communication Technologies, 2005. ICICT 2005. First International Conference on. 2005.
- [13] Rakesh, A. and K. Jerry, Watermarking relational databases, in Proceedings of the 28th international conference on Very Large Data Bases. 2002, VLDB Endowment: Hong Kong,
- [14] Sion, R., M. Atallah, and P. Sunil, Rights protection for relational data. *Knowledge and Data Engineering, IEEE Transactions on*, 2004. 16(12): p. 1509-1525.
- [15] Potdar, V.M., S. Han, and E. Chang. A survey of digital image watermarking techniques. in Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference on. 2005.
- [16] Radu, S., A. Mikhail, and P. Sunil, Resilient rights protection for sensor streams, in Proceedings of the Thirtieth international conference on Very large data bases - Volume 30. 2004, VLDB Endowment: Toronto, Canada.
- [17] Xiao, X., Sun, Xingming Lincong, Yang Minggang, Chen. Secure Data Transmission of Wireless Sensor Network Based on Information Hiding. in Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on. 2007.

- [18] Zhang, W., Liu, Y, Sajal K, Das Aggregation Supportive Authentication in Wireless Sensor Networks: A Watermark Based Approach. in World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a. 2007.
- [19] Xuejun, R., A sensitivity data communication protocol for WSN based on digital watermarking School of Information and Technology, Northwestern University, China, 2010.
- [20] Kamel, I.A.K., O. Al Dakkak, A. Distortion-Free Watermarking Scheme for Wireless Sensor Networks. in Intelligent Networking and Collaborative Systems, 2009. INCOS '09. International Conference on. 2009.
- [21] Haowen, C. and A. Perrig, Security and privacy in sensor networks. *Computer*, 2003. 36(10): p. 103-105.
- [22] Adrian, P.R., Szcwczyk J. D. Tygar Victor, Wen David, E. Culler, SPINS: security protocols for sensor networks. *Wirel. Netw.*, 2002. 8(5): p. 521-534.
- [23] Bartosz, P., S. Dawn, and P. Adrian, SIA: secure information aggregation in sensor networks, in Proceedings of the 1st international conference on Embedded networked sensor systems. 2003, ACM: Los Angeles, California, USA.
- [24] Xiaolong, L., et al., A Key Distribution Scheme Based on Public Key Cryptography for Sensor Networks, in Computational Intelligence and Security, W. Yuping, C. Yiu-Ming, and L. Hailin, Editors. 2007, Springer-Verlag. p. 725-732.
- [25] Yao, J., A security architecture for wireless sensor networks based-on public key cryptography 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing. 2009. 1.
- [26] Raghu, K.G.P., Jayachandran Haiyun, Luo Tarek, F. Abdelzaher, Datalink streaming in wireless sensor networks, in Proceedings of the 4th international conference on Embedded networked sensor systems. 2006, ACM: Boulder, Colorado, USA.
- [27] Shih, F., Zhang, M, Towards Supporting Contextual Privacy in Body Sensor Networks for Health Monitoring Service. W3C Workshop on Privacy and data usage control, 04/05 October 2010 Cambridge (MA).
- [28] E. Dawson, J.A., P. Gray, *Australasian Journal of Combinatorics* 1(1990), pp. 53-65.
- [29] Harjito, B., Watermarking Technique based on Linear Feed Back Shift Register (LFSR), Seminar Nasional Konferda ke -9 Himpunan Matematika Wilayah Jateng dan DIY di FMIPA UNS 2003.
- [30] O'connor, J., and Roberstson, E,F Andrew nikolaevich Kolmogorof school of mathematics and Statistics, university of St Andrews, Scotland, 1999.
- [31] Koushanfar, F. and M. Potkonjak, Watermarking Technique for Sensor Networks: Foundations and Applications. Book chapter, in 'Security in Sensor Networks', Yang Xiao (ed.), Auerbach publications 2006.
- [32] Wong, J.L., Feng, J, Kirovski, D, Potkonjack M, Security in sensor networks: watermarking techniques in Wireless sensor networks. 2004. 305.
- [33] Cox, I.J., Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 1997. 6(12): p. 1673.
- [34] Potdar, V., Sharif, A., and Chang, E., 2011. Industrial Strength Wireless Multimedia Sensor Network Technology. In: B. M. Wilamowski & J. D. Irwin eds. 2011. Industrial Electronics Handbook. 2nd ed. Boca Raton, FL, USA: CRC Press. Ch. 11. ISBN: 978-1-4398028-9-2.
- [35] Dillon, T., Talevski, A., Potdar, V., Chang, E., 2009. Web of things as a framework for ubiquitous intelligence and computing, In: Proceedings of the Ubiquitous Intelligence and Computing (UIC2009), Lecture Notes in Computer Science, vol. 5585, pp. 2-13.
- [36] Dillon, T., Potdar, V., Singh, J., Talevski, A., 2011. Cyber Physical Systems: Challenges in Sensor Actuator Networks, In: Proceedings of the 5th International Conference on Digital Ecosystems & Technologies (DEST2011), June 2011.
- [37] Rathnayaka, A.J.D., Potdar, V., 2011. A Critical Analysis of Wireless Sensor Network Transport Layer Protocol, *Journal of Network and Computer Applications*.
- [38] Sharif, A., Potdar, V., Chang, E., 2009. Wireless Multimedia Sensor Network Technology: A Survey. In: 7th IEEE International Conference on Industrial Informatics (INDIN 2009). Cardiff, UK, 2009, June 24-26.
- [39] Potdar, V., Han, S., and Chang, E., 2005. Fingerprinted Secret Sharing Steganography for Robustness against Image Cropping Attacks. In: Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN '05). Perth, Australia, August 10-12
- [40] Mohan, M., Potdar, V., and Chang, E., 2006. Recovering and Restoring Tampered RFID Data using Steganographic Principles. In: Proceeding of the 10th IEEE International Conference on Industrial Technology, (ICIT 2006). Mumbai, India, December 15-17.
- [41] Potdar, V., Chang, E., 2006. Tamper Detection in RFID Tags using Fragile Watermarking. In: Proceeding of the 10th IEEE International Conference on Industrial Technology. Mumbai, India, December 15-17.
- [42] Yousuf, Y. Potdar, V., 2008. A Survey of RFID Authentication Protocols. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW 2008). Okinawa, Japan, Mar 25 - 28.
- [43] Potdar, V., Hayati, P., and Chang, E., 2007. Improving RFID Read Rate Reliability by a Systematic Error Detection Approach. In: 1st Annual RFID Eurasia Conference. Istanbul, Turkey, September 5-6.
- [44] Han, S., Potdar, V., and Chang, E., 2007. Mutual Authentication Protocol for RFID Tags Based on Synchronized Secret Information with Monitor. In: 5th International Conference on Computational Science and Its Applications (ICCSA 2007), Malaysia, 26-29 Aug.
- [45] Potdar, V., Wu, C., Chang, E., 2005. Tamper detection for ubiquitous RFID enabled supply chain. In: Y. Hao et al. (Eds.); *Lecture Notes in Artificial Intelligence* 3802 Springer-Verlag Berlin Heidelberg 2005, vol. 3802, no. 1, page(s): 273-278, Dec. 2005.

- [46] Potdar, V., and Chang, E. 2004. Disguising Text Cryptography Using Image Cryptography. In Proc. of the 4th International Network Conference (INC 2004), Plymouth, U. K, July 6 -9.
- [47] B Harjito, V Potdar, J Singh, Watermarking technique for wireless multimedia sensor networks: a state of the art, Proceedings of the CUBE International Information Technology Conference, Pune, India, Sept 3-5, pp.832-840
- [48] Han, S., G. Skinner, Potdar, V., and Chang, E., 2006. A Framework of Authentication and Authorization for e-Health Services. In: Proceedings of the 3rd ACM workshop on secure web services (ACM SWS). Alexandria, Virginia, USA, Oct 27-31.

# Watermarking Technique for Wireless Multimedia Sensor Networks: A State of the Art

Bambang Harjito<sup>1,2</sup>

<sup>1</sup>School of Information Systems,  
Curtin University, Perth, Australia

<sup>2</sup>Informatic Department, FMIPA UNS  
Ir. Sutami No.36 A Surakarta 57126  
+62271663375 Indonesia  
harjito.bambang@postgrad.curtin.edu.au

Vidyasagar Potdar

School of Information Systems  
Curtin University

Perth, Western Australia  
v.potdar@curtin.edu.au

Jaipal Singh

Department of Computing  
Curtin University  
Perth, Australia

j.singh@curtin.edu.au

## ABSTRACT

Wireless multimedia sensor networks (WMSNs) are an emerging type of sensor network which contain sensor nodes equipped with microphones, cameras, and other sensors that produce multimedia content. These networks have the potential to enable a large class of applications ranging from military to modern healthcare. Multimedia nodes are susceptible to various types of attack, such as cropping, compression, or even physical capture and sensor replacement. Hence, security becomes an important issue in WMSNs. However, given the fact that sensors are resource constrained, the traditional intensive security algorithms are not well suited for WMSNs. This makes the traditional security techniques, based on data encryption, not very suitable for WMSNs. Watermarking techniques are usually computationally lightweight and do not require much memory resources. These techniques are being considered as an attractive alternative to the traditional techniques, because of their light resource requirements. The objective of this paper is to present a critical analysis of the existing state-of-the-art watermarking algorithms developed for WMSNs

## Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication. C.2.0 [General]: Security and protection. C.2.2 [Network Protocols]: Protocol architecture.

## General Terms

Algorithms, Performance, Design, Security.

## Keywords

Wireless sensor networks, wireless multimedia sensor networks and digital watermarking techniques

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) have the capability of sensing, processing, and wireless communication, all built into a tiny embedded device [19, 23, 32]. This type of network has attracted an increasing interest in the research community over the last few years. This interest is driven by theoretical and practical problems in embedded operating systems, network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CUBE 2012, September 3–5, 2012, Pune, Maharashtra, India.  
Copyright 2012 ACM 978-1-4503-1185-4/12/09...\$10.00.

protocols, wireless communications and distributed signal processing [24, 27, 30].

The primary function of WSNs is to collect and disseminate critical data that characterize the physical phenomena within the target area [25, 31]. Depending on the application scenario, WSNs can be categorized into two main streams: Wireless Scalar Sensor Networks (WSSNs) and Wireless Multimedia Sensor Networks (WMSNs) [1]. WSSNs are commonly called WSNs. The availability of low-cost cameras, CMOS image sensors and microphones, and also their broad application opportunities, including the ability to ubiquitously capture multimedia content from the environment, has fostered the development of WMSNs, i.e., the networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images, and scalar sensor data from the environment. Along with the ability to retrieve multimedia data, WMSNs also store, process in real-time, correlate and fuse multimedia data originating from heterogeneous sources [2, 26, 28]. WMSNs can not only change or enhance the existing sensor applications, such as tracking and environment monitoring [3], they can also enable several new applications, e.g., localization and recognition of services and users, control of manufacturing processes in industry [5], telemedicine, and attending to the disabled and elderly people by identifying the causes of the illnesses that affect them, such as dementia [4].

WMSNs have some novel features stemming from the fact that some sensor nodes have video cameras and higher computation capabilities. Consequently, WMSNs bring new opportunities as well as new challenges of security. Security is becoming an important issue with WMSNs. Due to the fact that WMSNs are vulnerable to different intentional network attacks, like man-in-the-middle attack [6], and also suffer from bad network channels [7], the authentication of the data transmitted cannot be verified. Man-in-the-middle attack can cause modification (insert, alter, delete) of the transmitted data, whereas bad network channels will introduce noise into the signal causing damage of data. Addressing these issues is important for a secure and trustworthy WMSN. However, the WMSN node has very limited power supply and computational capability, hence using a strong cryptographic algorithm with it becomes a challenge. Therefore, watermarking techniques are being investigated to address the issue of some of these attacks, like tempering, ownership etc [20, 22].

Hence, research in the area of watermarking and WMSN is becoming increasingly important [21, 29]. With the concept of the cyber physical system, i.e., the web of things, this research is

coming into the main stream and has become even more significant [33, 34]. In this paper, we investigate the current state-of-the-art technologies in the field of watermarking and the WMSNs.

The paper is structured as follows: in section 2, we provide an overview of WMSNs, section 3 overviews digital watermarking, section 4 describes digital watermarks in WMSNs, section 5 outlines an evaluation framework, section 6 describes the state of the art technologies for watermarking for WMSNs, and finally section 7 concludes the paper and indicates the lines for future work.

## 2. AN OVERVIEW OF WIRELESS MULTIMEDIA SENSOR NETWORKS

During the last few years, the availability of inexpensive CMOS cameras and microphones, coupled with the significant progress in distributed signal processing and multimedia source coding techniques, has made possible the development of WMSNs that are capable of gathering the multimedia information from the surrounding environment. WMSNs have deflected the main focus from the typical scalar WSNs to the networks with multimedia devices capable of retrieving video, audio, images, as well as scalar sensor data [8]. WMSNs are also able to deliver multimedia content. The general architecture of a multimedia sensor device may consist of several basic components, namely a sensing unit, a central processing unit, a communication subsystem, a coordination subsystem, a memory, and an optional mobility/actuation unit. It can be depicted as in Fig 1.

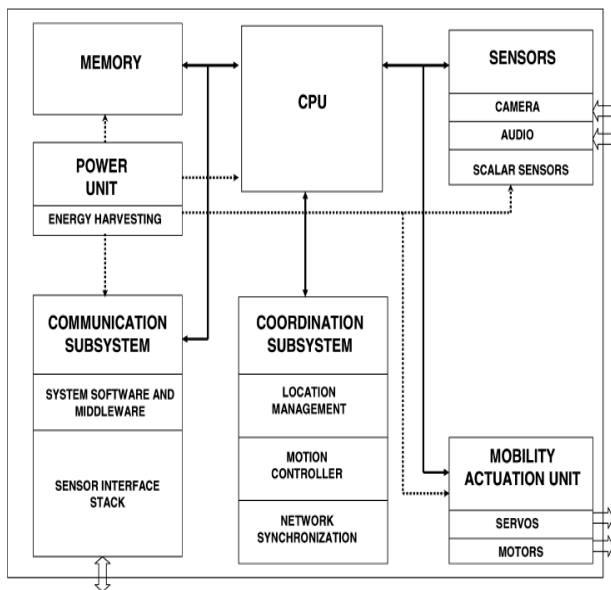


Figure 1. General hardware architecture of a multimedia sensor node

### 2.1 Sensing Unit

The sensing units are composed of two subunits: sensors (cameras, audio and/or scalar sensors) and analog-to-digital converters (ADCs). The camera and the audio sensors capture sounds, stills, moving images and the sensed events, and typically have resolutions in terms of pixel/inch for the camera

sensor, and in DB for the audio sensor. The scalar sensor senses the scalar data and the physical attributes, such as temperature, pressure, and humidity. The function of the ADCs is to convert the analog signals to digital signals. These analog signals are produced by the sensor, based on the observed phenomenon.

### 2.2 Central Processing Unit

Central Processing Unit (CPU) is the main controller of the multimedia sensor node. It executes the system software in charge of coordinating sensing and communication tasks, and this CPU is interfaced with a memory.

### 2.3 Memory Unit

The memory unit of the multimedia sensor node usually consists of both flash memory and RAM. The flash memory contains the programme code for the multimedia node, and the RAM stores information and any data required for computation. Some of the memory units also have non-volatile storage for off-line data capture for later retrieval.

### 2.4 Power Unit

The power unit is the most important component of the multimedia sensor node and is used to power the whole system. The power unit is supported by an energy scavenging unit, such as battery or solar cell.

### 2.5 Communication Subsystem

A communication subsystem interfaces the device to the network and is composed of a transceiver unit and the communication software. The communication software includes the communication protocol stack and the system software, for example, the operating system and the middleware.

### 2.6 Coordination Subsystem

A coordination subsystem is in charge of coordinating the operation of different network devices by performing operations, such as location management and motion control.

### 2.7 Mobility Actuation Unit

A Mobility actuation unit is optional in a multimedia sensor node. It can enable movement or manipulation of objects.

This concludes a brief description of WSN. We now provide a similar introduction to digital watermarking.

## 3. AN OVERVIEW OF DIGITAL WATERMARKING

Digital watermarking is the process of embedding information, which allows an individual to add hidden copyright notices or other verification messages to digital audio, video, or image signals and document objects [9-11]. Such hidden messages are groups of bits, describing information pertaining to the signal or the author of the signal. The signal may be audio, pictures or videos. If the signal is copied, the information is also carried in the copy. Watermarking seeks to embed a unique piece of data into the cover medium. The specific requirements of different watermarking techniques may vary with the applications, and there is no universal watermarking technique that would completely satisfy all the requirements for all applications.

The watermarking system as a communication task consists of three main stages: watermark generation process, watermark embedding process which includes information transmission



incase of possible attacks through the communication channels, and detecting process which consists of the watermark retrieval.

### 3.1 Watermark generator process

Watermark generation process is the first step in the watermarking system, and a very critical one. The requirements of the watermark generation process are unique and complex. The sensed data that a multimedia sensor node captures may be an image, an audio, a signal or a video. The watermark key is also unique in order to make a secrecy key, such as the threshold key [7] [12], weight coefficient [23], the user's insertion key [13] and the ID patient key [14]. Both the watermark message and the watermark key generator are used as inputs, and then are processed in the watermark generator to produce a watermark signal. Examples of watermark generator are the median filter [13], the 8-bit chirp signal [14], and the 5/8 encoder block [15]. The watermark signal is a kind of signal or pattern that can be embedded into the cover medium. There are two types of watermark signals, i.e., meaningful and meaningless watermarks. Examples of the meaningful watermarks are image logos, spread spectrum sequences, and permutations of the watermarks. On the other hand, pseudo-random sequences, binary matrices, M-sequences and chaotic sequences are examples of meaningless watermarks [16]. A generic digital watermarking system consists of the key components shown in Figure 2.

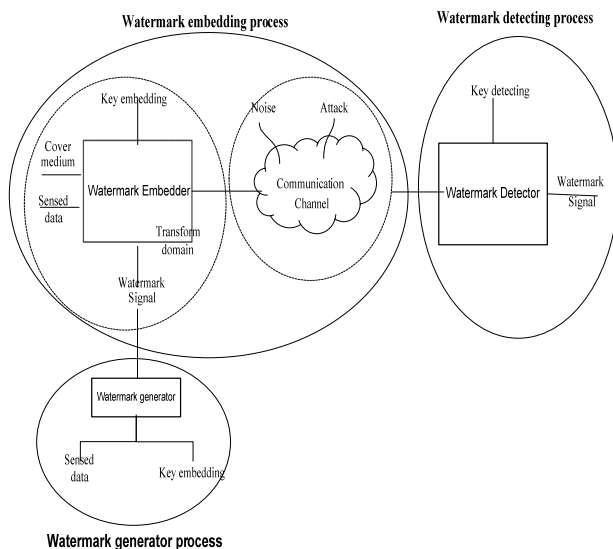


Figure 2. Digital Watermarking Process

### 3.2. Watermark embedding process

Embedding process is the second step in the watermarking system. This process is undertaken by an embedder, and can be done in the transform domain such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT) and Discrete Wavelet Transform (DWT). The embedder combines the cover medium, the watermark signal, the sensed data and the key embedding, and then creates the watermarked cover medium. Examples of the cover medium are packed data, texts, images, audio signals and videos. The watermarked cover medium is perceptually identical to the cover

medium, and is transmitted by the sender through unsecure communication channels, such as wireless and radio channels. During transmission, there are many things that may interfere in the communication process, such as noise, decreasing the quality of transmission and dropping the watermarked cover medium. The other threats are watermark attacks such as cropping, compression and filtering. The aim of these attacks is to remove the watermark signal from the watermarked cover medium.

### 3.3. Extracting & Detecting Process

The last stage of the watermarking system is the extraction or detection process which is a crucial part, as it enables the sender to identify and provide information to the intended receiver. The process of extraction or detection is undertaken by a detector. The detecting process consists of an extraction unit to first extract the watermark signal from the watermarked cover medium, and then compare it with original watermark signal from the cover medium. The extracting process can be divided into two phases, locating the watermark and recovering the watermark information. There are two types of detection: informed detection and blind detection, depending on whether the cover medium is required in the detection process or not. In case of informed detection, which involves the use of a cover medium, such as a packet data, original image or original signal, the watermarking system is called private watermarking. In case of blind detection, which does not need the cover medium detection, the watermarking system is called public watermarking.

This concludes the overview of digital watermarking, and we now move on to describing how WMSN and watermarking work together.

## 4. DIGITAL WATERMARKING TECHNIQUE IN WMSNs

In this section, we will now explain how WMSNs and digital watermarking technique can work together. We will show that digital watermarking technique can be implemented in WMSNs. This technique can also be used for implementing the specific name of the ownership in WMSNs. To accomplish the digital watermarking process, a typical encoder in WMSNs requires the original image which is obtained by the video sensor, and then this image is sent to the multimedia sensor node. The WMSNs, managed by the user, capture this image. Here, the watermark message is inserted into this image in order to prove the ownership of the content image. The process of embedding is as follows: first, the image is decomposed into several bands; then a pseudo-random sequence is added to the large coefficients which are not located in the lowest resolutions. The DWT watermark inserted algorithm consists of four parts, namely the original image, calculation of multilevel threshold, watermark embedding process, and inverse wavelet decomposition (IDWT). The watermarked image obtained by the embedding process is sent to the multimedia sensor node by the user. This image is then transmitted through a communication channel to a sink. The watermarked image is then managed again by the user who uses the laptop. The process of detecting or extracting is the inverse procedure of the watermark insertion process. It requires the watermarked image and the key.

This concludes the section on elementary concepts, and we now move on to an evaluation of different approaches to watermarking technique for WMSNs through a review of the literature in this field.

## 5. EVALUATION FRAMEWORK

To get an in-depth insight into the literature, we adopted an evaluation framework that critically analyses all the algorithms using the watermarking process, shown in Figure 3. We believe that this is the best approach to evaluate watermarking algorithms, because we can dissect the complete algorithm across different processes and components that form the overall watermarking process. From Figure 3, we can observe that there are three basic steps in watermarking process, as described in Section 3. We have further divided these three steps into eleven different components that form the part of the process. These include the following: (1) cover medium, (2) sensed data (3) watermark generator, (4) types of watermark, (5) watermark key, (6) watermarking embedding technique, (7) watermark detecting technique, (8) Attack, (9) noise, and (10) Transform domain

The main reason behind adopting this evaluation framework was to carry out an independent and thorough evaluation of each algorithm by clearly studying each watermarking aspect independently. In this study, we have selected six most recent and relevant algorithms published in the literature, which will now be evaluated in section 6.

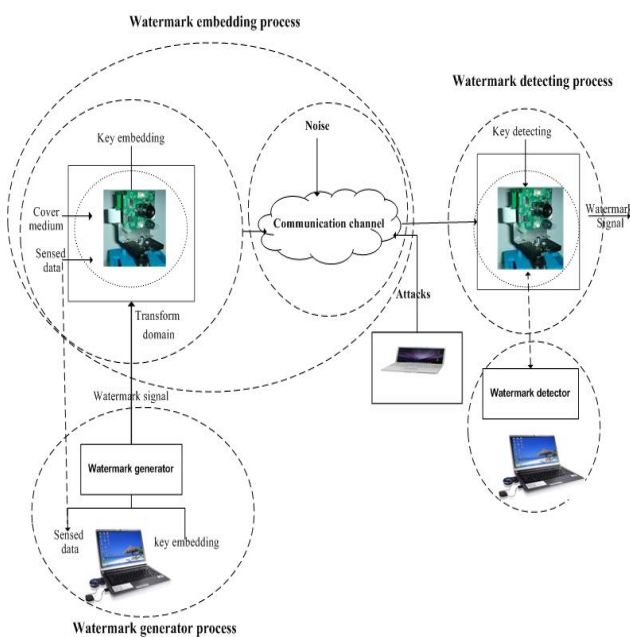


Figure 3. Digital watermarking for WMSNs

## 6. DIGITAL WATERMARKING TECHNIQUE FOR WMSNs : A STATE OF ART TECHNOLOGY

In this section we provide a detailed insight into the current literature on watermarking techniques for WMSNs. As described in section 5, we will now evaluate each algorithm by studying the ten components individually. We want to identify the similarities and differences in these algorithms, try to understand the rationale behind the authors' selection of a particular parameter for each component in their solution, and evaluate how good a choice it is.




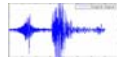


### 6.1. Watermark generator process

We begin the discussion with the first component of the watermark generation process, i.e. the sensed data.

#### 6.1.1. Sensed data

The sensed data, like the watermark message, has to be communicated through an insecure channel. However, in this case, a multimedia sensor node is better than a wireless sensor node, since it is capable of retrieving not only scalar data, but also video, audio, images and signals [8]. Here, the sensed data is generated by the multimedia sensor node. It can be interpreted as a copy right protection that has to be communicated to the other multimedia sensor node via an insecure channel. Different kinds of sensed data have been compared in all the different approaches, as presented in Table 1. As we found, the majority of these approaches have used 'image' as a sensed data [7], [17], [12], [15]. However, some of them have used 'signal' as a message [13], [14]. In this case, we cannot say which one is better, since it depends on the multimedia sensor node capture, whether image or signal.

Table 1. Sensed multimedia data used in WMSN literature

| Author                 | Year | Sensed Data                                                                                                    |
|------------------------|------|----------------------------------------------------------------------------------------------------------------|
| Honggang, et.al [7]    | 2008 | image<br>                   |
| Pingping et.al [17].   | 2009 | Image<br>                 |
| Wang et.al [12]        | 2010 | image<br>                 |
| Padmavathi, et al [13] | 2010 | audio acoustic signal<br> |
| Kaur, S et al [14]     | 2010 | ECG Signal<br>            |
| Masood, et al [15]     | 2011 | Image<br>                 |

#### 6.12 Key Embedding

The embedding process and the detecting process use a key which is called a watermark key whereby the watermark signal is inserted into the cover medium. The key is also used to enforce security, that is, to prevent an unauthorized party from recovering and manipulating the watermark. Here, we provide a comparative evaluation of the different types of watermark keys used in all these different approaches, as presented in Table 2. We found that 'the two adaptive threshold' has been used as a watermark

key in [7], [12], while some approaches have used a weight coefficient of the watermark [17], the user's insertion key [13], and the ID patient [14]. One of them does not mention the watermark key used [15]. We believe that the two adaptive threshold is better than a coefficient of the watermark as a watermark key, because it is used to filter and decide the appreciated embedding position [7]. On the other hand, the coefficient of the watermark cannot be used to filter and decide the appreciated watermark. However, we cannot exactly compare the two adaptive thresholds and the user ID, because they differ in their purposes.


**Table 2. Keys used in WMSN literature**

| Author                 | Year | Watermark key                              |
|------------------------|------|--------------------------------------------|
| Honggang, et.al [7]    | 2008 | the two adaptive threshold (Threshold key) |
| Pingping et.al [17].   | 2009 | Weight coefficient of the watermark signal |
| Wang et.al [12]        | 2010 | the two adaptive threshold (Threshold key) |
| Padmavathi, et al [13] | 2010 | The user's insertion key                   |
| Kaur, S et al [14]     | 2010 | ID patient Binary digit ( 15 bit)          |
| Masood, et al [15]     | 2011 | ---                                        |

### 6.13 Watermark Generator

Creating this type of watermark requires a watermark generator. We evaluated the generator used in all the different approaches presented in Table 3. The generator watermark for watermarking in WMSN consists of a Median filter, the 8-bit chirp signal and 5/8 encoder block. The majority of the authors do not mention what generator they have used [7] [17] [12]. Some of them have used the median filter [13], the 8-bit chirp signal [14] and 5/8 encoder block [15]. We again cannot say which kind of watermark generator is better because all three – the Median filter, the 8-bit chirp signal and 5/8 encoder block – are used for different purposes. The median filter generator produces a watermark signal in which the signal and the user key insertion are used as an input. While the watermark binary stream [15] is generated by the 8-bit chirp form, which signals an ID patient as an input, and the 5/8 encoder block generator produces a watermark signal from the image. Unfortunately, however, this generator can only use image as an input and cannot detect what watermark key has been used.




**Table 3 Watermark generator used in WMSN literature**

| Author                 | Year | Watermark generator                                                                                           |
|------------------------|------|---------------------------------------------------------------------------------------------------------------|
| Honggang, et.al [7]    | 2008 | ---                                                                                                           |
| Pingping et.al [17].   | 2009 | ---                                                                                                           |
| Wang et.al [12]        | 2010 | ---                                                                                                           |
| Padmavathi, et al [13] | 2010 | Median filter is used to denoise the signal                                                                   |
| Kaur, S et al [14]     | 2010 | The 8-bit chirp signal<br> |
| Masood, et al [15]     | 2011 | (5/8 Encoder Block)                                                                                           |

### 6.1.4 Watermark Signal

A pattern of bits is used as a watermark, and then the watermark is inserted into the cover medium. Examples of watermarks are image logos, binary matrices, audio data and signals. These watermarks can be inserted into a cover medium. We give a comparative evaluation of the different types of watermarks embedded into the cover medium implemented in WMSN. All these different approaches are presented in Table 4. We found that the majority of the authors have used 'signal' as a watermark [15] [13] [14], while some of them have used 'image logo' [7] [12]. Only a few of them have used the 'binary watermark' as a watermark signal [17]. We believe that the image logo is better than the binary matrix as a watermark signal, since the image logo can easily be detected and extracted. Using statistical approaches, such as NC, MSE and PSNR, the image logo can be detected [18] and its domain can be inverted, such as in IDWT, IFFT and IDCT. The image logo can also be separated from the cover medium, and then this logo can be seen with the human eye. In addition, the image logo is a meaningful type of watermark because people can still identify it through visual observation. On the other hand, the binary matrix is not commonly used. Signal as a watermark can be compared with image logo, because while the former has signal as the cover medium, the latter has image.

**Table 4. Watermarks used in WMSN literature**

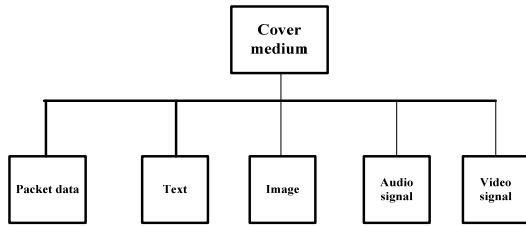
| Author                 | Year | Type of watermarks                                                                                                                                    |
|------------------------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Honggang, et.al [7]    | 2008 | Image logo<br>                                                   |
| Pingping et.al [17].   | 2009 | Binary matrix<br>$\begin{bmatrix} 1 & 0 & \dots & 1 \\ 1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 1 \end{bmatrix}$ |
| Wang et.al [12]        | 2010 | Image logo<br>                                                   |
| Padmavathi, et al [13] | 2010 | signal<br>                                                       |
| Kaur, S et al [14]     | 2010 | Binary stream                                                                                                                                         |
| Masood, et al [15]     | 2011 | Signal                                                                                                                                                |

## 6.2 Watermark embedding process

We begin the discussion with the first component of the watermark generation process, i.e. the cover medium.

### 6.2.1 Cover Medium

The cover medium is one of the key components of watermarking technique, and is used for inserting a watermark signal. There are different types of cover mediums, such as packed data, text, images, audio signals and videos, as presented in Figure 4.



**Figure 4. Cover medium**

From the investigated literature (Table 4), we provide a comparative evaluation of different cover mediums. As we found, the majority of them used ‘packed data’ as a cover medium [7] [17] [12] [15]. However, some of them used ‘audio signal’ [13], [14]. In this case again, we cannot say which one is better because their use depends on the main objective for inserting the watermark signal. For example Kaur et al [14] used a signal for inserting binary stream in which the binary stream was generated by an 8-bit chirp signal. This signal was used to protect the ECG signal from the patient. On the other hand, Honggang et al [7] and Wang et al [12] used packed data as the cover medium for embedding an image logo. The image logo was not produced by watermark generator, but was used to protect the cover medium. Here, both of them considered only the location of the image logo in the cover medium by using DWT.

**Table 5. Cover mediums used in WMSN literature**

| Author                 | Year | Cover medium |
|------------------------|------|--------------|
| Honggang, et.al [7]    | 2008 | Packet data  |
| Pingping et.al [17].   | 2009 | Packet data  |
| Wang et.al [12]        | 2010 | Packet data  |
| Padmavathi, et al [13] | 2010 | audio signal |
| Kaur, S et al [14]     | 2010 | audio signal |
| Masood, et al [15]     | 2011 | Packet data  |

### 6.2.2 Transform Domain

The transform domain of the digital watermarking technique can be divided into four categories, viz. Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), and Discrete Wavelet Transform (DWT). After investigating the literature (Table 6), we provide a comparative evaluation of the different types of transform domains. We found that the majority of them used ‘DWT’ as a transform domain [7] [12], while some of them used ‘DCT’ [17], [14] and ‘FFT’ [15]. However, one of them does not mention the transform domain used [13]. We believe that DWT is more robust than DCT and FFT because a watermark can be embedded into the selective coefficients at the three-level Discrete Wavelet Transform (DWT) middle frequency bands of an image frame, based on the network conditions.

**Table 6. The transform domains used in WMSN literature**

| Author                 | Year | Domain |
|------------------------|------|--------|
| Honggang, et.al [7]    | 2008 | DWT    |
| Pingping et.al [17].   | 2009 | DCT    |
| Wang et.al [12]        | 2010 | DWT    |
| Padmavathi, et al [13] | 2010 | ---    |
| Kaur, S et al [14]     | 2010 | DCT    |
| Masood, et al [15]     | 2011 | FFT    |

**Table 7. Watermarking techniques used in WMSN literature**

| Author                 | Year | Watermark embedding technique                                                                                                                                                                                                                                                                                                                                                         |
|------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Honggang, et.al [7]    | 2008 | $F_w$ as the watermarking function and $\{p_1, p_2, \dots, p_{mxw}\}$ is a set of the positions where the watermark is embedded under the specific watermarking schema. The two adaptive threshold $\{T_1, T_2\}$ is used to filter the appreciate embedding process. PLR is the packet lost ratio. So The function $D = F_{watermark}(\{p_1(T_1, T_2), \dots, p_i(T_1, T_2)\}, PLR)$ |
| Pingping et.al [17].   | 2009 | The embedding algorithm has the watermark equation $DCT'(p, q) = sign(DCT(p, q)) + W$ where $W$ is the watermark data, $\alpha$ is the weight coefficient of the watermark information and $(p, q)$ is location.                                                                                                                                                                      |
| Wang et.al [12]        | 2010 | The function $\{Q_w, Q_d\} = F_w(\{p_1(T_1, T_2), p_2(T_1, T_2), \dots, p_i(T_1, T_2)\}, PLR)$                                                                                                                                                                                                                                                                                        |
| Padmavathi, et al [13] | 2010 | The process of embedding digital data in the form of $X' = Ek(XW)$ , where $X$ is the pre-processed original signal, $W$ is the watermark information being embedded, $k$ is the user's insertion key                                                                                                                                                                                 |
| Kaur, S et al [14]     | 2010 | The function is $f_i(t) = f_o + \beta t^2$ where $\beta = (f_1 - f_0)t_1^{-2}$ . $y_{chirp, mod} = y_{chirp} * f(b_j)1$ $b_j$ is patient ID $y$ is the watermarked signal                                                                                                                                                                                                             |
| Masood, et al [15]     | 2011 | The embedding process is $d_w = E(d_o, k, w)$ . $d_o$ original $w$ watermark message $k$ security key                                                                                                                                                                                                                                                                                 |

### 6.2.3 Watermark Embedding Technique

Embedding a watermark signal is one part of the watermarking technique, the other is the process of detecting whether there is a watermark signal or not. We provide a comparative evaluation of the different kinds of watermarking techniques for WMSN.

While investigating the literature (Table 7), we found that two of the authors have used ‘the two filter adaptive threshold’ as an inserting technique [7] [12] in DWT, while others have used the weight coefficient of the watermark in DCT [17], the Orthogonal Frequency Davison Multiplexing (OFDM) in FFT [15], and The Wiener Filter [13] as the technique for embedding. We believe that ‘the two filter adaptive threshold’ is better as an inserting technique because it uses the three level DWT. Also, the adaptive threshold uses less power for WSN than the other techniques, because its positions are dynamically chosen to insert the watermark according to the network conditions, so that energy efficiency and security can be achieved [12].

#### 6.2.4 Noise

Noise can be defined as anything that influences the communication channel. The different types of noise are packet loss, decreasing the quality of transmission and packet drop. We provide a comparative evaluation of the different noise types for watermarking in WMSN. All these different approaches are presented in Table 8. As we found, the majority of these approaches have used ‘dropped packet data’ as the noise [7], [12, 13]. One of them has used ‘paper salt’ [17]. However, others have not mentioned what type of noise they have used [14, 15]. We believe that the drooped packet loss is more dangerous than decreasing the quality of transmission. Packet loss can stop the communication between the sender and the receiver because of lack of transmission. To overcome this noise, the sender retransmits the packet data. However, this transmission requires energy.

**Table 8. Example of Noise used in WMSN literature**

| Author                 | Year | Noise                 |
|------------------------|------|-----------------------|
| Honggang, et.al [7]    | 2008 | Packet loss           |
| Pingping et.al [17].   | 2009 | Paper salt noise      |
| Wang et.al [12]        | 2010 | Packet loss           |
| Padmavathi, et al [13] | 2010 | ---                   |
| Kaur, S et al [14]     | 2010 | any undesirable noise |
| Masood, et al [15]     | 2011 | Noise communication   |

#### 6.2.5 Vulnerable Attacks

There are two types of attacks: intentional attack and accidental attack. Intentional attacks include cryptanalysis, steganalysis, image processing techniques, and the removal of the existing watermark. Accidental attacks include the results of the standard image processing, such as filtering, resizing or the compression procedure. The different attacks used for watermarking technique in WMSN belong to the category of accidental attack, such as cropping, compression, and filtering. Here, we provide a comparative evaluation of the different vulnerable attacks. All these different approaches have been presented in Table 9. We found that the majority of these approaches have used the ‘accidental’ type as the vulnerable attack, such as cropping and compressing [7], [12], [17], while one of them has used ‘filtering’ [14]. However, some of them have not mentioned the type of attack used for their watermarking technique [13]. We believe that cropping and compressing are more possible attacks than filtering, since these are accidental attacks.

**Table 9. Watermark Attacks used in WMSN literature**

| Author                 | Year | Noise                    |
|------------------------|------|--------------------------|
| Honggang, et.al [7]    | 2008 | Compression              |
| Pingping et.al [17].   | 2009 | Cropping and Compression |
| Wang et.al [12]        | 2010 | Compression              |
| Padmavathi, et al [13] | 2010 | ---                      |
| Kaur, S et al [14]     | 2010 | Filtering                |
| Masood, et al [15]     | 2011 | ---                      |

**Table 10. Watermark detection techniques used in WMSN literature**

| Author                 | Year | Watermark extracting technique                                                                                                                                     |
|------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Honggang, et.al [7]    | 2008 | To detect the watermark image, the normalized correlation (NC) coefficient to measure similarity of original watermarks and extracted watermark                    |
| Pingping et.al [17].   | 2009 | Peak signal-to Noise ratio (PSNR) and the extracted watermark is obtained by comparing d with 0, i.e., $W' = \begin{cases} w'=0 & d=0 \\ w'=0 & d>0 \end{cases}$ . |
| Wang et.al [12]        | 2010 | Normalized correlation (NC)<br>$NC = \frac{\sum_{i=1}^w \sum_{j=1}^m w(i,j) \cdot w^*(i,j)}{\sum_{i=1}^w \sum_{j=1}^m [w(i,j)]^2}$                                 |
| Padmavathi, et al [13] | 2010 | Mean Square Error (MSE) and Peak signal-to Noise ratio (PSNR)                                                                                                      |
| Kaur, S et al [14]     | 2010 | ---                                                                                                                                                                |
| Masood, et al [15]     | 2011 | The watermark detection process is defined as $W_e = D(d_w, k, w, d_o)$ watermarked data<br>Do original data<br>watermark w;<br>Key k;                             |

### 6.3. Watermark Detecting Process

The detecting process consists of an extraction unit to first extract the watermark signal from the watermarked cover medium, and then compare it with original watermark signal from the cover medium. The process of extracting or detecting is used to check whether there is a watermark signal or not in the cover medium. Here, we provide a comparative evaluation of this process used for watermarking in WMSN in the literature we surveyed. All the different approaches are presented in Table 10. As we found, the majority of these approaches have used the ‘statistic approach’ as the detecting technique, such as Normalized Correlation (NC) and Peak Signal-to-Noise Ratio (PSNR) [7] [12], [17], [13]. However, some of them have used ‘The 8-bit chirp signal’ as their detecting process. We believe that the statistic approach is better than the 8-bit chirp signal because it is more common and

valid method used without the medium signal. Use of the medium signal for detection is impossible in WMSN, because the watermark image is invisible to the eye.

## 7. CONCLUSION

Using strong cryptography algorithms, such as RSA, ECC and digital signature are not suitable for WSN and WMSN because these algorithms are prohibitively expensive in terms of energy and storage requirements. Watermarking techniques are being investigated to address some of these network issues, such as tempering, deleting and manipulating data packet. These techniques are much lighter and require less battery power and processing capabilities than cryptography-based algorithms. In addition, the advantage of these techniques is that the watermark signal is embedded directly into the sensor data, so that there is no increase in the payload. In this paper, we surveyed and evaluated 6 current approaches used in the existing literature for watermarking technique in WMSN, using 10 different parameters. Based on this evaluation of literature on digital watermarking technique for WMSNs, we have provided a summary of the majority of these approaches using each of the parameters for digital watermarking technique for WMSNs.

## 8. ACKNOWLEDGMENTS

We thank ACM SIGCHI for allowing us to modify the templates they had developed. We also sincerely thank all the CUBE conference program committee members and anonymous reviewers for providing in-depth reviews and constructive feedback, which helped us to improve this manuscript significantly.

## 9. REFERENCES

- [1] Manel Guerrero Zapata, R.Z., Jos'e M, Barcel'o-Ordinas, Kemal Bicakci, Bulent Tavli, *The Future of Security in Wireless Multimedia Sensor Networks*. 2009.
- [2] Akyildiz, I.F., T. Melodia, and K.R. Chowdhury, *A survey on wireless multimedia sensor networks*. Computer Networks, 2007. **51**(4): p. 921-960.
- [3] Jason, C., Phillip, B. Gibbons, Suman, Nath, Padmanabhan, Pillai Srinivasan, Seshan Rahul, Sukthankar, *IrisNet: an internet-scale architecture for multimedia sensors*, in *Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, ACM: Hilton, Singapore.
- [4] Reeves, A.A., *Remote monitoring of patients suffering from early symptoms of dementia*. in Proc. Int. Workshop Wearable Implantable Body Sensor Networks: p. London, U.K., Apr. 2005.
- [5] Bisma, R.A., Nash, R. Aragam, Yi, Yao Mongi, A. Abidi, *Survey and analysis of multimodal sensor planning and integration for wide area surveillance*. ACM Comput. Surv., 2008. **41**(1): p. 1-36.
- [6] Kamel, I., *A Lightweight Data Integrity Scheme for Sensor Networks*. Sensors, 2011. **11**(4): p. 4118.
- [7] Honggang, W.D., Peng Wei, Wang Sharif, H. Hsiao-Hwa, Chen. *Energy-Aware Adaptive Watermarking for Real-Time Image Delivery in Wireless Sensor Networks*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008.
- [8] Islam T. Almalkawi, M.G.Z., Jamal N. Al-Karaki, Julian Morillo-Pozo, *Wireless Multimedia Sensor Networks: Current Trends and Future Directions*. Sensors, 2010. **10** p. 6662 - 6717.
- [9] Wang, X.-Y., Z.-H. Xu, and H.-Y. Yang, *A robust image watermarking algorithm using SVR detection*. Expert Systems with Applications, 2009. **36**(5): p. 9056-9064.
- [10] Potdar, V., *Subjective and Objective Watermark Detection Using a Novel Approach—Barcode Watermarking* ed. C.I.a. Security. 2007. 576.
- [11] Vidyasagar, P., J. Christopher, and C. Elizabeth, *Multiple image watermarking using the SILE approach*, in *Proceedings of the 6th WSEAS international conference on Multimedia systems signal processing*. 2006, World Scientific and Engineering Academy and Society (WSEAS): Hangzhou, China.
- [12] Wang, H., *Communication-resource-aware adaptive watermarking for multimedia authentication in wireless multimedia sensor networks*. The Journal of Supercomputing, 2010: p. 1-15.
- [13] Padmavathi, G., D. Shanmugapriya, and M. Kalaivani. *Digital watermarking technique in vehicle identification using wireless sensor Networks*. in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*.
- [14] Kaur, S., *Digital Watermarking of ECG Data for Secure Wireless Communication International Conference on Recent Trends in Information, Telecommunication and Computing*. 2010 140.
- [15] Masood, H.H., U. Sadiq ur, Rehman Khosa, I. *Secure communication in WMSN*. in *Information Networking and Automation (ICINA), 2010 International Conference on*. 2010.
- [16] Bai, B., J. Harms, and Y. Li, *Configurable active multicast congestion control*. Computer Networks, 2008. **52**(7): p. 1410-1432.
- [17] Pingping, Y.S., Yao Jiangtao, Xu Yu, Zhang Ye, Chang. *Copyright Protection for Digital Image in Wireless Sensor Network*. in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*. 2009.
- [18] Wenjun, Z. and B. Liu, *A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images*. *Image Processing, IEEE Transactions on*, 1999. **8**(11): p. 1534-1548.
- [19] Potdar, V., Sharif, A., and Chang, E., 2011. Industrial Strength Wireless Multimedia Sensor Network Technology. In: B. M. Wilamowski & J. D. Irwin eds. 2011. *Industrial Electronics Handbook*. 2nd ed. Boca Raton, FL, USA: CRC Press. Ch. 11. ISBN: 978-1-4398028-9-2.
- [20] Potdar V., Chang, E., 2006. Tamper Detection in RFID Tags using Fragile Watermarking. In: Proceeding of the *10th IEEE International Conference on Industrial Technology*. Mumbai, India, December 15-17.
- [21] Mohan, M., Potdar, V., and Chang, E., 2006. Recovering and Restoring Tampered RFID Data using Steganographic Principles. In: Proceeding of the *10th IEEE International Conference on Industrial Technology*, (ICIT 2006). Mumbai, India, December 15-17.
- [22] Potdar, V., Han, S., and Chang, E., 2005. A Survey of Digital Image Watermarking Techniques. In: *Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN '05)*. Perth, Australia.
- [23] Harjito, B., Han, S., Potdar, V., Chang, E., Ma, X., 2010. Secure Communication in Wireless Multimedia sensor

- Networks using watermarking, In: *International Conference on Digital Ecosystems and Technologies* (IEEE DEST 2010). Dubai, U.A.E.
- [24] Potdar, V., Sharif, A., and Chang, E., 2009. Wireless Sensor Networks: A Survey. In: *2nd International Workshop on RFID and its Industrial Applications*, 23rd International Conference on Advanced Information Networking and Applications Workshops (WAINA '09). Bradford, UK, May 26-29.
- [25] Sharif, A., Potdar, V., Rathnayaka, A. J. D., 2010. LCART: Lightweight Congestion Aware Reliable Transport Protocol. *Australian Journal of Intelligent Information Processing Systems*, 12(1), pp. 1-9.
- [26] Rathnayaka, A.J.D., Potdar, V., 2011. A Critical Analysis of Wireless Sensor Network Transport Layer Protocol, *Journal of Network and Computer Applications*.
- [27] Sharif, A., Potdar, V., Rathnayaka, A. J. D., 2010. ERCTP: End-to-End Reliable and Congestion Aware Transport Layer Protocol for Heterogeneous WSN. *Special issue of Journal of Scalable Computing: Practice and Experience*, 11(4), pp. 359–371.
- [28] Sharif, A., Potdar, V., Rathnayaka, A. J. D., 2010. Dependency of Transport Functions on IEEE802.11 and IEEE802.15.4 MAC/PHY Layer Protocols for WSN: A Step towards Cross-layer Design. *International Journal of Business Data Communications and Networking*, 6(3), pp. 1-30.
- [29] Potdar, V., Han, S., Chang, E., Wu, C., 2007. Subjective and Objective Watermark Detection using a Novel Approach - Bar-code Watermarking. *Lecture Notes in Artificial Intelligence*, 4456(1), pp. 576-586.
- [30] Sharif, A., Potdar, V., Chang, E., 2009. Wireless Multimedia Sensor Network Technology: A Survey. In: *7th IEEE International Conference on Industrial Informatics* (INDIN 2009). Cardiff, UK, 2009, June 24-26.
- [31] Sharif, A., Potdar, V., and Rathnayaka, A. J. D., 2010. LCART: Lightweight Congestion Aware Reliable Transport Protocol for WSN Targeting Heterogeneous Traffic. In: *17th International Conference on Neural Information Processing* (ICONIP 2010). Sydney, Australia, Nov 22 - 25.
- [32] Rathnayaka, A. J. D., Potdar, V., Sharif, A., 2010. Wireless Sensor Network Transport Protocol: A State of the Art. In: *1st International Workshop on Wireless Sensor Networks, Wireless Multimedia Sensor Networks & RFID* (WSNR). Japan, November.
- [33] Dillon, T., Talevski, A., Potdar, V., Chang, E., 2009. Web of things as a framework for ubiquitous intelligence and computing, In: *Proceedings of the Ubiquitous Intelligence and Computing* (UIC2009), Lecture Notes in Computer Science, vol. 5585, pp. 2-13.
- [34] Dillon, T., Potdar, V., Singh, J., Talevski, A., 2011. Cyber Physical Systems: Challenges in Sensor Actuator Networks, In: *Proceedings of the 5th International Conference on Digital Ecosystems & Technologies* (DEST2011), June 2011.

# Watermarking Technique for Wireless Sensor Networks: A State of the Art

Bambang Harjito<sup>1,2</sup>, Vidyasagar Potdar<sup>1</sup>, Jaipal Singh<sup>3</sup>

<sup>1</sup>*School of Information Systems, Curtin University, Perth, Australia*

<sup>2</sup>*Informatic Department, Faculty of Mathematics and Natural Science. Sebelas Maret University, Surakarta, Indonesia*

<sup>3</sup>*Department of Computing, Curtin University, Perth, Australia*

[harjito.bambang@postgrad.curtin.edu.au](mailto:harjito.bambang@postgrad.curtin.edu.au) [v.potdar@curtin.edu.au](mailto:v.potdar@curtin.edu.au) [j.singh@curtin.edu.au](mailto:j.singh@curtin.edu.au)

**Abstract**— Wireless sensor networks (WSN) are used in a number of different applications including military, environment monitoring, smart spaces etc. Security is an important aspect in WSN, however given scarce resources; implementing traditional cryptographic algorithms is not very well suited for WSN nodes. Hence watermarking is being investigated as an alternative security technology in WSNs because of its light resource requirements. The objective of this paper is to present a critical analysis of the existing state-of-the-art watermarking algorithms developed for WSNs.

## Keywords

Cyber-physical systems (CPS), Wireless Sensor Networks and Digital Watermarking.

## I. INTRODUCTION

Wireless sensor networks (WSN) is one of the key technologies in connecting the physical and the cyber world as sensors monitor the physical environment and transmit the physical state information over a network to a database for further processing [1, 2, 16, 17, 18]. Such systems are referred to as Cyber-physical-systems (CPS). Security in CPS is very important. However, sensor in CPS are very low powered and ensuring security becomes challenging [1] [2]. Hence watermarking techniques is being considered as an alternative approach to offers some degree of security such as tamper detection, content ownership data authentication. Watermarking process involves four main stages: watermark generation, watermark embedding, communication channel and watermark detecting and retrieval. This paper investigates the current state-of-the-art in the field of watermarking for WSN. The paper is structured as follows: section 2 compares different aspects parameter for watermarking technique WSN, and finally we have concluded and future works the paper in section 3.

## II. WATERMARKING TECHNIQUE FOR WSNs: A STATE-OF-THE-ART

In this section we provide detailed insight into the current literature on watermarking techniques for wireless sensor networks [16, 17, 18].

### A. Watermark Generation Process

Watermark generation process involves three main components i.e. watermark message, watermark key and the watermark generator

#### 1. Watermark Message

Watermark message is a secret message embedded in a cover medium. In the context of watermarking for WSN, it would most likely be plain text, as using image as a watermark would drain the sensor resources very fast. When investigating the literature (Table 1), we found that majority of the approaches do not specifically mention what they used as a watermark message in their experiments [3], [4], [5] [6], [7], [8], [9]. However Feng [10] & Koushanfar [11] suggested using “text” as a watermark message [10], [11]. We believe that “text as a watermark message” is better than using a random text because text based watermark will at least produce a legible signature which will be helpful during the detection process, especially in subjective detection approaches.

#### 2. Watermark key

Watermark key is a secret or public key used in embedding and the detecting process. From the investigated literature (Table 1), we found that the two different types of keys are used as a watermark key, these include (1) pseudorandom sequence, (2) binary stream. We found that that majority of the approaches used “binary stream” as the watermark key such as [3] [5] [7] [8] [9]. However, some of them suggested using a pseudorandom sequence [6] as watermark key. However others such as [10, 12],[11], [4] did not mention what kind of watermark keys they used in their studies. We believe that binary stream is better than using a pseudorandom sequence  $s(x,y)$  because the binary stream use less energy and can be easily added to a hash function to produce a watermark signal. Whereas the pseudorandom sequence  $s(x,y)$  needs high cost computationally and use more energy than the binary stream.



TABLE 1 STATE OF THE ART COMPARISON OF DIFFERENT WSN WATERMARKING ALGORITHMS

| Author                  | Watermark Message | Watermark Key                  | Watermark generator                                | Cover medium | Sensed data   | Type of watermark       |
|-------------------------|-------------------|--------------------------------|----------------------------------------------------|--------------|---------------|-------------------------|
| Feng et al. [10]        | Plain text        | ---                            | H, (MD5), RSA, RC4                                 | NLSP         | ---           | Watermark constraints   |
| Sion et al. [4]         | ---               | ---                            | H                                                  | P            | B             | B                       |
| Koushanfar et al. [11]  | Plain text        | ---                            | H, (MD5), RSA, RC4                                 | NLPS         | ---           | Watermark constraints   |
| Julia Albath et al. [5] | ---               | B                              | H                                                  | P            | B             | B                       |
| Zhang et al. [6]        | ---               | Pseudorandom sequence $s(x,y)$ | Product function of $b_i, \alpha(x,y)$ and $(x,y)$ | P            | Matrix binary | Watermark bits $w(x,y)$ |
| Juma et al. [7]         | ---               | B                              | H                                                  | P            | B             | B                       |
| Xiao et al. [8]         | ---               | B                              | H                                                  | P            | B             | B                       |
| Xiaomei et al. [9]      | ---               | B                              | H                                                  | P            | B             | B                       |
| Xuejun et al. [12]      | ---               | ---                            | ---                                                | P            | B             | B                       |
| Ren et al. [13]         | ---               | ---                            | H                                                  | P            | B             | B                       |
| Kamel et al. [3]        | ---               | B                              | H                                                  | P            | B             | B                       |

B= Binary stream ((101010 ... ), H = Hash function and P = Packet data

### 3. Watermark Generator

Watermark generator is a mathematical function which is used for generating watermark signal using watermark message and watermark key. Two types of watermark generators are found in the WSN watermarking literature and they are (1) hash function and (2) product function. When investigating the literature (Table 1), we found that majority of these approaches used “hash function” as a watermark generator, such as MD5 and SHA [3] [10] [11] [5] [7] [9]. Only one of them [6] used “product function of  $b_i \in \{-1,1\}, \alpha(x,y)$  and  $s(x,y)$ ” to generate a watermark signal. However, others did not mention what they used for generating the watermark signal [8] [12]. We believe that hash function is better than product function because the hash function can be defined as deterministic procedure which takes arbitrary block of data such as binary stream and returns a fixed-size bit string. The binary stream is encoded by the hash function to become the hash value which is called the message digest. It uses less energy. Whereas product function of  $b_i \in \{-1,1\}, \alpha(x,y)$  and  $s(x,y)$  needs high cost computationally than the hash function.

#### B. Watermark Embedding Process

Watermark embedding is the second stage in the overall watermarking process. It includes:

##### 1. Cover medium

Cover medium refers to packet data, text, image, audio signal and video signal, as well as a Non Linear Programming System ( a NLPS ). When investigating the literature (Table 1), we found that majority of these approaches use “packet data” as a cover medium [3] [4] [5] [6] [7] [8] [9] [12] [13]. However some of them have used the NLPS as a cover medium [10] [11]. We believe that packet data is better than the NLPS because the packet

data is a basic unit of communication over a digital network. To transmit the whole data, the whole data is broken into small chunks called packets [14]. With the packet data, each of sensor nodes only consumes network resources when they are actually transferring data. The power consumption and the networks resource utilization are suitable for wireless sensor nodes because of their energy constraints. While the NLPS approach does not provide any energy consumption studies in their paper [10]

##### 2. Sensed Data

Sensed data refers to the external environment phenomenon captured by sensors such as temperature, humidity, speed, direction, movement, light etc. In WSN this is the crucial information which has to be secured as well can communicated to the sink. When investigating the literature (Table 1), we found that majority of these approaches use “binary stream ” as a sensed data [3], [4], [7], [8], [9], [12], [13]. One of them suggested using “matrix binary “ as sensed data [6]. However some of them do not mention what they used as a sensed data [5], [10], [11]. We believe that binary stream as sensed data better than matrix binary because the binary stream is a small chunk of the whole data and need less power energy when it transmits by wireless sensor node [14]. While the matrix binary is a pseudorandom code which is generated serving as a modulation pulse to spread the signal across the entire band [6]. This matrix binary does not mention how much it needs energy power.

##### 3. Watermark Signal

Watermark signal is the actual copyright information that is embedded in the cover medium. Three different types of watermarks are used in WSNs (1) watermark constraint, (2) binary stream and (3) watermark bits.

TABLE 2: WATERMARK EMBEDDING TECHNIQUES, WATERMARK DETECTION, ATTACKS AND NOISE USED IN WSN

| Author                  | Watermark embedding technique                                                                                                                                  | Watermark detection technique                                     | Attacks                                                                                               | Noise                                     |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Feng et al. [10]        | Adding watermark constraint to processing step during network operation                                                                                        | ---                                                               | ---                                                                                                   | ---                                       |
| Sion et al. [4]         | Selection criteria using MSB                                                                                                                                   | Similarity                                                        | Sampling, Segmentation, Linear changes, addition stream                                               | Incoming data rate                        |
| Koushanfar et al. [11]  | Adding watermark constraint to processing step during network operation                                                                                        | ---                                                               | Ghost Signature, Removal of the author's signature, De-synchronization                                | ---                                       |
| Julia Albath et al. [5] | generating the one-time pad by repeatedly concatenating the substring                                                                                          | Calculate MAC over p(xi) and compare to MAC received with packet. | ---                                                                                                   | ---                                       |
| Zhang et al. [6]        | The watermark sensory data ,<br>$d(x, y) = w(x, y) + o(x, y)$<br>$w(x,y)$ is the watermark for sensor node and<br>$O(x,y)$ is sensory data                     | A Gaussian of hypothesis testing on correlation coefficient       | False distribution imposition all sensor , False distribution imposition part of sensor Remnant check | Node failure                              |
| Juma et al.[7]          | Concatenation of the current group hash value group $gi$ and next group hash value group $gi+1$ . $W = \text{HASH} (K    gi    gi+1)    gi$                    | synchronization point or not                                      | Modify data<br>Adding data false                                                                      | --                                        |
| Xiao et al. [8]         | By modification the embedding bit of each packet. LSB                                                                                                          | Manipulation embedding bit and decrypt with key                   | ---                                                                                                   | ---                                       |
| Xiaomei et al. [9]      | The random value of each send data and time was calculated by inputting the collection time and its MSB and the key K into random function.                    | Mean and standard deviation                                       | Personate , Forgery, Sampling, Summarization                                                          | Decrease the quality of transmitting data |
| Xuejun et al. [12]      | IIS = input integer stream, IBS=input binary stream. T = Threshold, If IIS $\geq$ T "IBS=1" become "IBS=0"<br>Else "IBS=0" become "IBS=0"                      | ---                                                               | ---                                                                                                   | ----                                      |
| Ren et al. [13]         | Embed bit of watermark by changing the parity of its LSB.                                                                                                      | By judging the parity of the LSB of each data.                    | Selective forward, Data replication , Packet transfer delay, Packet tampering                         | ---                                       |
| Kamel et al. [3]        | Concatenation of the current group hash value group $gi$ and next group hash value group $gi+1$ . $Wi = \text{HASH} (K    gi    SN) SN = \text{serial number}$ | Comparison hash calculated between sender and receiver            | Data modification attack<br>False data insertion<br>Data deletion                                     | ---                                       |

We found that majority of them used “binary stream” as a watermark signal (8, 16, 64, 128 bits) [3] [4] [5][7] [8] [9] [12] [13]. Two of them [10] [11] used a watermark constraint. Only one of them suggesting used a watermark bit  $w(x,y)$ . We believe that watermark binary stream is better than using watermark constraints or watermark bit because the watermark binary stream can be produced by calculating the watermark bit of the MSB which denotes the most significant bit of hash function [5] [8] [9]. In addition this watermark binary stream can also be computed by hash function which is applied to the concatenation of all individual data elements in the group [3, 7]. Generate the binary stream using the hash function use less energy and computationally more efficient than watermark constraints and watermark bits because they need high cost energy and computationally less efficient than the binary stream.

#### 4. Watermark embedding technique

Watermarking embedding refers to the process of adding a watermark signal to the cover media. Here, we give critical evaluation of different embedding techniques

proposed for WSNs. Table 2 documents these techniques. We found that majority of them use “Least Significant Bit (LSB)” as a embedding technique [3-9, 12, 13]. However some of them use adding watermark constraint to processing step during network operation [10] [11]. We believe that LSB is much better because LSB is the most straight-forward method of watermark embedding and suits very well especially for low powered wireless sensor nodes. LSB technique uses a simple replace operation to embed watermark signal in cover medium. Given the high channel capacity of using entire cover medium for transmission in this method, a watermark signal may be embedded multiple times. Even if most of these are lost due to attack, a single surviving watermark signal would be sufficient to prove.

### C. Communication Channel

#### 1. Noise

Anything that inferences in the communication channel between the sender and the receiver is called noise. There can be different types of noise in communication channel

in WSN such as incoming data rate, probability of collisions, node failure and dropped packet data etc. We found that majority of them do not mention the nature of noise [3] [7] [8] [12] [13] [10] [11]. Some of them explain that incoming quality data rate [4], dropped packet data [5, 6], decrease the quality transmitting data [9] as a noise.

## 2. Vulnerable attacks

During data transmission there is a possibility of attacks which can destroy the watermark. Table 2 provides a comparative evaluation of different attack experienced by WSN. We found that majority of them use “man-in-middle attack” as vulnerable attack such as modification, false data insertion, forgery, personate[3] [4], [7] [9] [13] Ghost signature, de-synchronization [11], statistic attack [6, 9, 13] are used as vulnerable attack. However some of them do not mention their attack [10] [5] [8] [12] We believe that man-in-middle attack is more likely than statistic attack because this attack is commonly used.

## D. Watermark Detection Process

### 1. Watermark detecting technique

The detection process is used for deciding the presence of a watermark in the cover medium. Table 2 provides details about the detection techniques. We found that majority of them use “statistic correlation” as a detection technique such as similarity, probability, Gaussian hypothesis, mean and standard deviation [4] [5] [6] [8] [9] [12]. However some of them suggest using synchronization point [7], manipulation embedding bit [8] and comparison hash calculated between sender and receiver [3]. However some of them do not mention what they have used for detection [10] [11] [12] [13]. We believe that statistic is better than manipulation embedding bit and comparison hash calculated between sender and receiver because statistic is a valid method for detecting watermark without using the original cover medium from a wireless sensor node as a sender [15]. Generally, in watermark detection, given the test cover medium assumed to have been marked and whose ownership is to be determined, one first extracts the watermark signal from, usually making use of the original cover medium. Then it is compared to the original watermark signal signature. The similarity index is then compared to a threshold to determine if the test cover medium is a watermarked version of the original cover medium [15]. However, using the original cover medium to detect is unrealistic and not suitable for WSN.

## III. CONCLUSION AND FUTURE WORK

In this paper we presented an in depth evaluation of the state of the art watermarking techniques for securing wireless sensor networks. We studied 11 different algorithms in this study and compared it against each other across 10 different dimensions, which represent the key components and processes of digital watermarking. This study was undertaken to identify research gaps in the

literature and to develop new watermarking technique for wireless sensor networks. The future work involves developing a watermarking algorithm for WSN..

## REFERENCES

1. Dillon, T., Potdar, V., Singh, J., *Cyber-physical systems: Providing Quality of Service (QoS) in a heterogeneous systems-of-systems environment*. in *Digital Ecosystems and Technologies Conference (DEST), 2011 Proceedings of the 5th IEEE International Conference on*.
2. Dillon, T., Talevski Alex, Potdar Vidyasagar, Chang, Elizabeth, Zhang, Daqing, Portmann, Marius, Tan, Ah-Hwee, Indulska, Jadwiga, *Web of Things as a Framework for Ubiquitous Intelligence and Computing Ubiquitous Intelligence and Computing*. 2009, Springer Berlin / Heidelberg. p. 2-13.
3. Kamel, I., *A Lightweight Data Integrity Scheme for Sensor Networks*. *Sensors*, 2011. **11**(4): p. 4118.
4. Radu, S., A. Mikhail, and P. Sunil, *Resilient rights protection for sensor streams*, in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*. 2004, VLDB Endowment: Toronto, Canada.
5. Albath, J., *Practical algorithm for data security (PADS) in wireless sensor networks* *Proceedings of the 6th CM international workshop on Data engineering for wireless and mobile access - MobiDE '07*. 2007. 9.
6. Zhang, W., Liu, Y, Sajal K, Das *Aggregation Supportive Authentication in Wireless Sensor Networks: A Watermark Based Approach*. in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. 2007.
7. Juma, H.K., I.Kaya, L. *Watermarking sensor data for protecting the integrity*. in *Innovations in Information Technology, 2008. IIT 2008. International Conference on*. 2008.
8. Rong, X., S. Xingming, and Y. Ying. *Copyright Protection in Wireless Sensor Networks by Watermarking*. in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHMSP '08 International Conference on*. 2008.
9. Xiaomei, D.X., Li. *An Authentication Method for Self Nodes Based on Watermarking in Wireless Sensor Networks*. in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*. 2009.
10. Fang Jessica, P., Miodrag *Real-time watermarking techniques for sensor networks* *Proceedings-SPIE The International Society for optical Engineering* 2003(ISSU 5020): p. 391-402
11. F. Koushanfar, M.P., *Watermarking Technique for Sensor Networks: Foundations and Applications*. Book chapter, in ‘Security in Sensor Networks’, Yang Xiao, Auerbach publications, 2007.
12. Xuejun, R., *A sensitivity data communication protocol for WSN based on digital watermarking* School of Information and Technology, Northwestern University, Xi’an 710127, China, 2010.
13. Ren, B.W.X.S.Z.R.H., *Multi-mark: Multiple Watermarking Method for Privacy Data Protection in Wireless Sensor Networks*. *Information Technology Journal*, 2011. **10**(4): p. 833-840.
14. Wuyungerile, L., M. Bandai, and T. Watanabe. *Tradeoffs among Delay, Energy and Accuracy of Partial Data Aggregation in Wireless Sensor Networks*. in *AINA 2010, Perth, Australia*.
15. Wenjun, Z. and B. Liu, *A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images*. *Image Processing, IEEE Transactions on*, 1999. **8**(11): p. 1534-1548.
16. V. Potdar, E. Chang, “A Survey of Digital Image Watermarking techniques”, *INDIN 2005, Perth, Australia*.
17. V. Potdar, A. Sharif, E. Chang, “Wireless sensor networks: A survey”, in *Proceedings of the IEEE AINA 2009, 636- 641, Perth, Australia, 2009*.
18. V. Potdar, E. Chang, “Tamper detection in RFID tags using fragile watermarking”, in *Proceedings of IEEE ICIT 2006, 2846- 2852, Mumbai, India, 2006*.

## Wireless Multimedia Sensor Networks Applications and Security Challenges

Bambang Harjito<sup>1,2</sup>

<sup>1</sup>Digital Ecosystem and Business Intelligence  
Institute  
Curtin University of Technology, Perth, Western  
Australia

[harjito.bambang@postgrad.curtin.edu.au](mailto:harjito.bambang@postgrad.curtin.edu.au)

<sup>2</sup>Computer Science Department, Faculty of  
Mathematics and Natural Science  
Sebelas Maret University, Surakarta, Indonesia

Song Han<sup>1</sup>,

<sup>1</sup>Digital Ecosystem and Business Intelligence  
Institute  
Curtin University of Technology, Perth,  
Western Australia

[Song.Han@cbs.curtin.edu.au](mailto:Song.Han@cbs.curtin.edu.au)

**Abstract—** The emergence of low-cost and mature technologies in wireless communication, visual sensor devices, and digital signal processing facilitate of wireless multimedia sensor networks (WMSNs). Like sensor networks which respond to sensory information such as humidity and temperature, WMSN interconnects autonomous devices for capturing and processing video and audio sensory information. WMSNs will enable new applications such as multimedia surveillance, traffic enforcement and control systems, advanced health care delivery, structural health monitoring, and industrial process control. Due to WMSNs have some novel features which stem the fact that some of the sensor node will have video cameras and higher computation capabilities. Consequently, the WMSNs bring new security of challenges as well as new opportunities. This paper presents WMSNs application and security challenges.

**Keywords-component; Security, Wireless Multimedia Sensor Networks.**

### I. INTRODUCTION

The availability of multimedia devices such as a small microphones and low-cost complementary metal-oxide semiconductor (CMOS) has fostered the development of wireless multimedia sensor network (WMSN). These multimedia devices can capture multimedia content such as scalar data, stream audio and video from the environment. Thereby a WMSN will have the ability to transmit and to receive multimedia information such as monitoring data, image, voice, and stream video. Since the ability to retrieve multimedia information so the WMSN will also be able to store, process in real time, correlate and fuse multimedia information from different sources. Thus, WMSNs are composed of numerous type multimedia sensors which exchange sensed multimedia data with sink by using wireless channel [1]. WMSNs will not only change enhance existing sensor applications such as tracking, and environment monitoring, but they will also enable several new applications. For example they range over systems supporting telemedicine to modern military.

In WMSNs, data harvested from the environmental is not only a scalar nature which is obtained from various internal sensor such as temperature, light, humidity, pressure, and acoustic

sensor but also from multimedia data such as digital images, video and audio form [2]. Therefore the main sensor in WMSN is the imager. The visual data which is handled puts severe constraints on a sensor network. Collection, processing, and visual data dissemination is a processing intensive and high bandwidth demanding operation. WMSNs have some novel features which stem the fact that some of the sensor node will have video cameras and higher computation capabilities. Consequently, the WMSNs bring new security of challenges as well as well as some new opportunities.

The paper is structured as follows: Section 2 explains WMSNs components, Section 3. present review all the aspects of WMSNs architecture Section 4 describes WMSNs application, Section 5 describe security and challenges, Section 6 we examine a number of security design and finally we have concluded the paper in section 7.

### II. WIRELESS MULTIMEDIA SENSOR NETWORK HARDWARE COMPONENTS

In this section we discuss hardware components of WMSN specially in the multimedia sensor node. In [2], it states that an enabling hardware platforms multimedia sensor hardware has divided in two categories depending on it's resolution. Low-resolution imaging motes and Medium-resolution imaging motes based on the Star gate platform [2]

- The low-resolution imaging motes.

The technology of CMOS imaging sensor that capture and process an optical image allows integrating a lens, an image sensor and image processing algorithms, including image stabilization and image compression, on a single chip. Existing CMOS images is still developed. In [3] introduced a Cyclops which consist of an image CMOS Agilent ADCM-1700 CIF camera), a complex programmable logic device, an external SRAM and an external flash. The objective of this Cyclops is to fill a gap between computational devices and CMOS cameras. The design of an integrated mote with wireless sensor networks is explained in [4]. This design is based on an adequate processing power and memory size for application. Here a new image mote is based on an ARM7 32-bit CPU clocked at 48 MHz, with external FRAM or

Flash memory, 802.15.4 compliant Chipcon CC2420 radio and low-resolution 30 x 30 pixel optical sensors.

- Medium-resolution imaging motes based on the Star gate platform

The stargate board [5] was produced by Crossbow and designed by Intel. This stargate board, based on Intel's PXA-255 Xscale 400MHz RISC processor, is a high performance processing platform designed for sensor, robotic and sensor networks applications. It also 32 Mbyte of flash memory, 64 Mbyte of SDRAM and on-board connector Crossbow's MICAz or MICA2. The another prototype has developed by Intel such as Imote and Imote2.

Suh et al [6] introduced a novel solution for improving IEEE 802.15.4 performance with the adaptive active duration via two data traffic indication schemes, designed and implemented a real sensor platform and its camera module for testbed experiments. The development of a low cost, low power WSN hardware platform named *TelG* embedded with an operating system called *WiseOS*, system software, and also a simple best effort JPEG images transmission over the network [7]

There has been a lot of work to develop in this field however the growing technology is still not mature and still need several technical challenges.

### III. WIRELESS MULTIMEDIA SENSOR NETWORK ARCHITECTURE

In this section we survey the network architecture for WMSN in [1, 2, 8, 9]. The basic architecture of WMSN. It can be shown in Fig.1

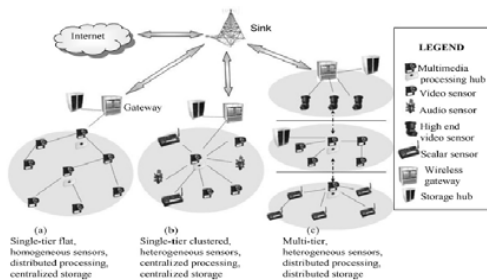


Fig. 1 Architecture of WMSN [2]

There are three model depending on the targeting application nature i.e., the first model is a single-tier flat homogeneous (multi-hop) camera sensor network where the sink is a wireless gateway connected to a centralized storage hub, the second reference model is a single-tiered clustered network with heterogeneous sensors. Camera, audio and scalar sensors relay data to a cluster-head able to perform intensive data processing and the cluster-head is connected to a gateway as in the first model and the third model is a multi-tier architecture with heterogeneous sensors. In the first tier, scalar sensors perform simple tasks, like motion detection. A second tier of camera sensors perform more complicated tasks such as object detection or object recognition. At the end point, high end video sensors are connected to wireless gateways.

Elhadi et al [8] was described typical characteristics of WMSNs and introduced a reference architecture for WMSNs similar to [2]. It can be show in Fig. 2.a. Grieco et al [1] introduces an architecture of WMSN. It is composed of numerous multimedia sensors that exchange sensed data with sinks using a wireless channel. It can be seen in Fig.2.b.



Fig. 2 Architecture of WMSN [8] and [1]

There have been many designs to develop an architecture for WMSNs. The design is still similar to [2].

### IV. WIRELESS MULTIMEDIA SENSOR NETWORK APPLICATIONS

The characteristic of a WMSN diverge consistently from traditional network paradigms, such as the internet and even from the WSNs. The most potential applications of WMSNs require the sensor networks paradigm to be rethought to provide mechanisms to delivery multimedia contents with the predetermined level of Quality of Service (QoS). WMSNs will enable several new applications.

- *Surveillance*: WMSNs are used currently in surveillance which need streaming multimedia content, advanced signal and high bandwidth. Such as Audio and video sensors will be used to complement and enhance existing surveillance systems against crime attack [10].
- *Traffic monitoring and Enforcement* : WMSNs are low cost, easy of deployment and ease for reconfiguring routes when deployment in the specific location such as in big cities of highways. They will be possible to monitor car traffic and to service that offer traffic routing advice to avoid congestion [11].
- *Personal and Health care* : WMSNs, incorporated with some telemedicine devices, can be used to remotely monitor the patient's body temperature, blood and breathing activity etc. They can be studied the behavior of elderly people as means to identify the causes of illnesses that affect them such as dementia [12].
- *Gaming*: WMSNs will find applications in the future prototypes that enhance the effect on the game player. Such as virtual reality games that assimilate touch and sight input, of user as part of the player response [13].
- *Environmental and industrial* : Array of video sensors are used by Oceanographer to determine the evolution of sandbars using image processing

techniques [14] and multimedia content such as imaging, temperature, or pressure can be used for time-critical industrial process control.

## V. SECURITY CHALLENGES IN WIRELESS MULTIMEDIA SENSOR NETWORKS

In this section we discuss security and challenge for WMSNs. In WMSN, we know that data harvested from the target area is not only scalar nature such as humidity, temperature, light, pressure, seismic but also more complex such as image and streaming video. Therefore the energy dissipation in multimedia sensor nodes is dominated by the computation energy rather than the communication energy.

Manel et al [9] provided a survey and analysis of the different security issues that will have to take into account in the design of WMSNs platforms and protocol. Grieco et al [1] summarize the main findings on secure WMSNs and forecasts future perspectives of such a technology. Here we address security according to these into four categories.

### 1. Efficient management of Quality of Experience (QoE) and Quality of Service (QoS).

The requirement of the multimedia monitoring applications state new problems which wireless communication and infrastructures of processing have to solve to assure the QoE. Such as the problem of the limited power resources and computational capabilities [15, 16] and the problem of computational complexity of compression are considered. In addition the problem of aggregation, distributed processing, the overload to manage privacy/security and QoS are also considered.

### 2. Privacy

In WMSNs collect and handle a great amount of data of different nature, which may provide some kind of information on individuals in both a direct or indirect form. The kind of information may specify explicit information on individuals. Therefore, under some circumstances, data may be used to violate the privacy of individuals. Privacy is a key requirement for numerous application scenarios of WMSNs. The privacy solutions, such as secure data cloaking [17], secure communication channel [18, 19] and definition of privacy policies [20-22] are not enough to provide a complete privacy solution for WMSNs. Due to each solution satisfies only specific requirements ad-hoc problems.

### 3. Authentication and Node localization

In WSNs communications, Han et al [23] describe a taxonomy of attacks on WSNs. The taxonomy consists of six attacks i.e., communication attacks, attacks against privacy, sensor node targeted attacks, power consumption attacks, policy attacks and cryptology attacks on key management. In communication attacks, Eavesdropping can easily inject messages, so the receiver needs to make sure that the data used any decision-making

process originates from the correct source. Data authentication prevents unauthorized parties from participating in the networks and legitimate nodes should be able to detect messages from authorized nodes and reject them. The authenticity of these data and commands is a critical requirement for the correct behaviour of a WMSN [1]. The problem of the authentication is strictly related to the secure node localization issue [23]. Authentication can be used to ensure reliable information. Because of the distributed nature of WMSNs, the localization of the multimedia sensors is required to assure the supply of the services. Therefore, the integrity and confidentiality of localization information are fundamental and it is necessary to define countermeasures versus possible malicious attacks.

### 4. Development of Platform.

The integration of existing and upcoming solutions, such as aggregation algorithms, compression technique, secure localization, authentication mechanisms, should be allowed by the platform such as sensEye [24-26]. Here, the platform considers QoE, security, privacy and technological constraints. The reference platform should be hierarchical. Each level of the hierarchical could be use different protocol/algorithm and technologies. Research efforts should be a opportunity to foster new collaborations between different academic and industrial group.

## VI. DESIGN CHALLENGES FOR SECURITY SCHEMES WIRELESS MULTIMEDIA SENSOR NETWORKS

In this section, we give security design challenge for WMSNs. Sensor nodes are often deployed in unattended and even harsh environments. They may suffer from many kinds of attacks. Wireless channels are low-cost and unreliable. The transmission of data packets may be delay or may not reach its destination. Indeed, security challenges and opportunities in WMSNs stem from these characteristics [9]. Here we examine a number of security design challenges:

### • Unattended deployment environments:

Sensor nodes often deployed in a large unattended area. An attacker may compromise one or a number of sensor nodes without being noticed. As a consequence, no solution is specifically deployed for WMSNs. Hence, new approaches exploiting the characteristics of multimedia nodes should be developed [1]. There are many papers that explain to deploy sensor node. Such as designing multimedia sensor networks to support volcanic studies requires addressing the high data rates and high data fidelity and sparse array with high spatial separation between nodes [27], however in this paper is not described the security of the sensor node. Tzu et al [28] explains

a procedure of deployment for a wireless sensor network. It is addressed to guide users to complete the deployment tasks systematically and Younis et al [29] survey on the current state of the research on optimized node placement in WSNs. Both of them is only used to deploy WSN and is not concerned to secure of the sensor node.

- **Data Privacy:**

Privacy issues are of concern in WSNs, if the collected data is private and sensitive. Video, image and audio data are typically more sensitive than scalar data, such as temperature. Hence, privacy enhancing techniques, such as source location, hiding and distributed visual secret-sharing [18, 19] may be crucial for WMSNs. Attacks versus privacy which exploit these vulnerabilities can be categories into distinct macro-types of techniques: Eavesdropping and Masquerading. The design of privacy protecting mechanisms is a challenging problem for the intrinsic characteristic of WMSNs.

Gruteser et al [30] proposed a methodology for identifying, assessing and comparing location privacy risk in mobile computing technologies. However, this method cannot be used for design securing in WMSNs. The source location privacy problem is studied in [31] under the assumption of one single source during a specific period. However, this method is not specifically defined for WMSNs.

Yi et al [32] propose a Proxy based Filtering Scheme (PFS) and a Tree-based Filtering Scheme (TFS), which are simple yet efficient event source unobservability preserving solutions for sensor networks. However these methods are not suitable for securing in WMSNs.

- **Data authentication:**

Wireless communications make security and privacy requirements critical because they increase the vulnerabilities and the threats on the integrity and confidentiality of the transmitted data. For these reasons, authentication mechanisms [33] are required to guarantee the correctness and the confidentiality of data. Moreover, due to the high number of sensor nodes, such systems could contain control units that broadcast commands and data to the nodes. Hence, the authenticity of these data and commands is a critical requirement for the correct behaviour of WMSNs. Data authentication guarantees and ensures that raw data are received at the aggregators at the same time as they are being sensed. Zhang et al [34] proposed a watermark statistical approach for data authentication in WSN which provides inherent support for in-network processing. The data authentication is only work from sensor nodes to the data sink. However secure data authentication is not explained from the sink to the sensor node. In the literature [35, 36] provide authentication algorithm for data authentication however this algorithm is not

adequately satisfy the quality of service requirements of multimedia signals.

- **Multimedia in-network processing :**

Multimedia in networks processing is one of the factors influencing the design of WMSNs. WMSNs allow algorithm of processing of multimedia content from the environment. A new architecture for collaborative, distributed, and resource-constrained processing is required. This architecture allows for filtering and extraction of semantically relevant information at the edge of the sensor network. Nath et al [37] introduces IrisNet which uses application-specific filtering of sensor feeds at the source and reduces the bandwidth consumed, since instead of transferring raw data, IrisNet sends only a potentially small amount of processed data. Stockdon, et al [38] introduce distributed filtering technique that can create a time-elapsd image in video security application. Both of them is concerned to specific filtering of sensor. However they do not concerned for securing in networks processing in WMSNs.

## VII. CONCLUSION

In this paper, we aims to address the problem of secure and challenges in WMSNs . We have also discussed the existence hardware component and surveyed the network architecture for WMSNs. The application of WMSNs is explained. This paper is also figure out a number of security challenges. Based on this paper, we will next to try to design a conceptual frame work for securing in WMSNs.

## REFERENCES

1. Grieco, L.A., Boggia, G, Sicari, S, Colombo, P. *Secure Wireless Multimedia Sensor Networks: A Survey*. in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBIKOMM '09. Third International Conference on*. 2009.
2. Akyildiz, I.F., T. Melodia, and K.R. Chowdhury, *A survey on wireless multimedia sensor networks*. *Computer Networks*, 2007. **51**(4): p. 921-960.
3. Mohammad, R.R., Baer Obimdinachi, I. Iroez Juan, C. Garcia Jay, Warrior Deborah, Estrin Mani, Srivastava, *Cyclops: in situ image sensing and interpretation in wireless sensor networks*, in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. 2005, ACM: San Diego, California, USA.
4. Ian Dowes, L.B., Hmid Aghajan *Development of a Mote for Wireless Sensor Networks*. in. *Proceeding of Cognitive System and Interactive Sensor(COGIS)* Paris, , 2006.
5. Crossbow Mote Specifications, h.w.x.c., <http://www.xbow.com>, 3th July 2010
6. Suh, C.M., Zeeshan Hameed Ko, Young-Bae, *Design and implementation of enhanced IEEE 802.15.4 for supporting multimedia service in Wireless Sensor Networks*. *Computer Networks*, 2008. **52**(13): p. 2568-2581.
7. Abdul Hadi Fikri Bin Abdul Hamid, R.A.R., Norsheila Faisal, S. K. S. Yusof, S. H. S. Ariffin Liza Latiff, *Development of IEEE802.15.4 based Wireless Sensor Network Platform for Image Transmission*. *International Journal of Engineering & Technology IJET* 2009. **9**(10): p. 7.
8. Elhadi Shakshuki, X.X., Haroon Malik *An Introduction to Wireless Multimedia Sensor Networks*. 2009: p. 16.

9. Manel Guerrero Zapata, R.Z., Jos'e M. Barcel'o-Ordinas, Kemal Bicakci, Bulent Tavli, *The Future of Security in Wireless Multimedia Sensor Networks*. 2009.
10. Dan, L.W., K. D. Yu Hen, Hu Sayeed, A. M., *Detection, classification, and tracking of targets*. Signal Processing Magazine, IEEE, 2002. **19**(2): p. 17-29.
11. Arth, C.B., H. Leistner, C. TRICam - *An Embedded Platform for Remote Traffic Surveillance*. in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*. 2006.
12. Reeves, A.A., *Remote monitoring of patients suffering from early symptoms of dementia*. in Proc. Int. Workshop Wearable Implantable Body Sensor Networks: p. London, U.K., Apr. 2005.
13. Mauricio, C., et al., *The multimedia challenges raised by pervasive games*, in *Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, ACM: Hilton, Singapore.
14. Rob, H., S. John, and O.-H. Tuba, *Applying Video Sensor Networks to Nearshore Environment Monitoring*. IEEE Pervasive Computing, 2003. **2**(4): p. 14-21.
15. I. Downes, L.B.R., H. Aghajan, *Development of a mote for wireless image sensor networks*. in Proc. of COGNITIVE systems with Interactive Sensors, COGIS Paris, France: p. Mar. 2006.
16. Margi, C.B., et al. *Characterizing energy consumption in a visual sensor network testbed*. in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*. 2006.
17. Kundur, D.L., W. Okoroafor, U. N. Zourntos, T. *Security and Privacy for Distributed Multimedia Sensor Networks*. Proceedings of the IEEE, 2008. **96**(1): p. 112-130.
18. Douglas, A.F., N. Hoang-Anh, and T. Mohan, *The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks*, in *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. 2004, ACM: New York, NY, USA.
19. Adrian, P.R., Szewczyk J. D. Tygar Victor, Wen David, E. Culler, *SPINS: security protocols for sensor networks*. Wirel. Netw., 2002. **8**(5): p. 521-534.
20. Sastry, D.M., Gruteser Xuan, Liu Paul, Moskowitz Ronald, Perez Moninder, SinghJung-Mu, Tang, *Framework for security and privacy in automotive telematics*, in *Proceedings of the 2nd international workshop on Mobile commerce*. 2002, ACM: Atlanta, Georgia, USA.
21. Qun, N., Alberto, Trombetta, Elisa, Bertino, Jorge, Lobo, *Privacy-aware role based access control*, in *Proceedings of the 12th ACM symposium on Access control models and technologies*. 2007, ACM: Sophia Antipolis, France.
22. Mark, M.J., rg, Schwenk, *Security model and framework for information aggregation in sensor networks*. ACM Trans. Sen. Netw., 2009. **5**(2): p. 1-28.
23. Han S , L.G., Chang E , Tharam D, *Taxonomy of Attacks on Wireless Sensor Networks*. Proceedings of the First European Conference on Computer Network Defence School of Computing, University of Glamorgan Wales, UK, 2006.
24. Wu-Chi, F.E., Kaiser Wu Chang, Feng Mikael Le, Baillif, *Panoptes: scalable low-power video sensor networking technologies*. ACM Trans. Multimedia Comput. Commun. Appl., 2005. **1**(2): p. 151-167.
25. Teresa, A.D.A., Nasipuri Craig, Taylor, *Explorebots: a mobile network experimentation testbed*, in *Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*. 2005, ACM: Philadelphia, Pennsylvania, USA.
26. Purushottam, K.D., Ganesan Prashant, Shenoy Qifeng, Lu, *SensEye: a multi-tier camera sensor network*, in *Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, ACM: Hilton, Singapore.
27. Geoffrey, W.-A.K., Lorincz Matt, Welsh Omar, Marcillo Jeff, Johnson Mario, Ruiz Jonathan, Lees, *Deploying a Wireless Sensor Network on an Active Volcano*. IEEE Internet Computing, 2006. **10**(2): p. 18-25.
28. Tzu-Che Huang, H.-R.L.a.C.-H.K., *A deployment procedure for wireless sensor networks*. Networks and Multimedia Institute, Institute for Information Industry., 2007.
29. Younis, M.A., Kemal, *Strategies and techniques for node placement in wireless sensor networks: A survey*. Ad Hoc Networks, 2008. **6**(4): p. 621-655.
30. Marco Gruteser, D.G., *A Methodological Assessment of Location Privacy Risks in Wireless Hotspot Networks*. In First International Conference on Security in Pervasive Computing, 2003.
31. Y.Xi, L.S., and W.Shi, W, *Preserving source location privacy in monitoring-based wireless sensor networks*. Parallel and Distributed Processing Symposium 2006. IPDPS 2006. 20th International 2006: p. 8.
32. Yi, Y.M., Shao Sencun, Zhu Bhuvan, Urgaonkar Guohong, Cao, *Towards event source unobservability with minimum network traffic in sensor networks*, in *Proceedings of the first ACM conference on Wireless network security*. 2008, ACM: Alexandria, VA, USA.
33. Honggang, W.D., Peng Wei, Wang Sharif, H. Hsiao-Hwa, Chen. *Energy-Aware Adaptive Watermarking for Real-Time Image Delivery in Wireless Sensor Networks*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008.
34. Zhang, W.L., Yonghe Das, Sajal K. De, Pradip, *Secure data aggregation in wireless sensor networks: A watermark based authentication supportive approach*. Pervasive and Mobile Computing, 2008. **4**(5): p. 658-680.
35. Liu, D. and P. Ning, *Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks*. 2002, North Carolina State University at Raleigh.
36. Liu, D. and P. Ning, *Multi-Level microTESLA: A Broadcast Authentication System for Distributed Sensor Networks*. 2003, North Carolina State University at Raleigh.
37. S. Nath, Y.K., P.B. Gibbons, B. Karp, S. Seshan, *A distributed filtering architecture for multimedia sensors*. Intel Research Technical Report IRP-TR-04-16, agustus, 2004.
38. H. Stockdon, R.H., *Estimation of wave phase speed and nearshore bathymetry from video imagery*. J. Geophys Res. **105** (C9) 22,015-22,033, 2000.



# Secure Communication in Wireless Multimedia Sensor Networks using Watermarking

Bambang Harjito<sup>1,2</sup>, Song Han<sup>1</sup>, Vidyasagar Potdar<sup>1</sup>, Elizabeth Chang<sup>1</sup>, Miao Xie<sup>1</sup>

<sup>1</sup>Digital Ecosystem and Business Intelligence Institute  
Curtin University of Technology, Perth, Western Australia

[harjito.bambang@student.curtin.edu.au](mailto:harjito.bambang@student.curtin.edu.au)  
{[Song.Han](mailto:Song.Han@curtin.edu.au); [Vidyasagar.Potdar](mailto:Vidyasagar.Potdar@curtin.edu.au); [Elizabeth.Chang](mailto:Elizabeth.Chang@curtin.edu.au)}@cbs.curtin.edu.au  
[Miao.X@curtin.edu.au](mailto:Miao.X@curtin.edu.au)

<sup>2</sup>Computer Science Department, Faculty of Mathematics and Natural Science  
Sebelas Maret University, Surakarta, Indonesia  
[bambangcs@mipa.uns.ac.id](mailto:bambangcs@mipa.uns.ac.id)

**Abstract-** *Wireless multimedia sensor networks (WMSNs) are an emerging type of sensor networks which contain sensor nodes equipped with microphones, cameras, and other sensors that producing multimedia content. These networks have the potential to enable a large class of applications ranging from military to modern healthcare. Since in WMSNs information is multimedia by nature and it uses wireless link as mode of communication so this posse's serious security threat to this network. Thereby, the security mechanisms to protect WMSNs communication have found importance lately. However given the fact that WMSN nodes are resources constrained, so the traditionally intensive security algorithm is not well suited for WMSNs. Hence in this research, we aim to a develop lightweight digital watermarking enabled techniques as a security approach to ensure secure wireless communication. Finally aim is to provide a secure communication framework for WMSNs by developing new.*

**Index terms** - *Wireless Multimedia Sensor Networks, Watermark, Digital watermarking*

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) have the capability for sensing, processing and wireless communication all built into a tiny embedded device. This type of network has drawn increasing interest in the research community over the last few years. This is driven by theoretical and practical problems in embedded operating systems, network protocols, wireless communications and distributed signal processing. The primary function of WSNs is to collect and disseminate critical data that characterize the physical phenomena within the target area. Depending on the application scenario WSNs can be categorized into two main streams: Wireless Scalar Sensor Networks (WSSNs) and Wireless Multimedia Sensor Networks (WMSNs) [1]. In addition, The availability of low-cost cameras, CMOS image sensor and microphones, also their broad application opportunities that are able to ubiquitously capture multimedia content from the environment has fostered the development of WMSNs, i.e., networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images, and scalar sensor data from the environment. To the ability to retrieve multimedia data, WMSNs will also be able to store, process in real-time,

correlate and fuse multimedia data originated from heterogeneous sources. WMSNs will not only change enhance existing sensor applications such as tracking, and environment monitoring [2], but they also will enable several new applications. For example they range over systems supporting telemedicine, attendance to disabled and elderly people as means to identify the causes of illnesses that affect them such as dementia [3], localization and recognition of services and users, and control of manufacturing processes in industry [4].

WMSNs have some novel features which stem the fact that some of the sensor node will have video cameras and higher computation capabilities. Consequently, the WMSNs bring new security of challenges as well as new opportunities. Security is a key concern in such application like traffic monitoring and enforcement [2] and monitoring process in industry [4]. However given the problems including the limited power resources and computational capabilities, it is difficult to implement strong cryptography algorithm. Hence, this paper aims to investigate the possibility of digital watermarking technique as an alternative method for providing security. The paper is structured as follows : Section 2 present review all the aspects of secure WMSNs, Section 3 describes proposed framework for watermarking enabled secure communication in WMSNs, Section 4 describe framework implementation, Section 5 evaluation and finally we have concluded and future work the paper in section 6.

## 2. RELATED WORKS

WMSNs security is still a very young research field. Tavli et al [1] provided a survey and analysis of the different security issues that will have to take into account in the design of WMSNs platforms and protocol. Grieco et al [5] summarize the main findings on secure WMSNs and forecasts future perspectives of such a technology. Both of them will be spurred new research ideas. Here, we integrate prior research results and investigate the following sub categories: which includes privacy, authentication

mechanisms, secure communication channels and Secure Compression and aggregation of multimedia data contents.

### 2.1 Privacy

In WMSNs collect and handle a great amount of data of different nature, which may provide some kind of information on individuals in both an indirect or direct form. The kind of information may specify explicit information on individuals. Therefore, under some circumstances, data may be used to violate the privacy of individuals. Privacy is a key requirement for numerous application scenarios of WMSN[5]. WMSNs run the risk of individual privacy violation due to possible unauthorized access to the data that are handled by the network. This treat is mainly attributable to vulnerabilities of WMSN, for example the remote access data and the huge quantity of multimedia data that are exchanged within the network [5]. Attacks versus privacy which exploit these vulnerabilities can be categories into distinct macro-types of techniques: Eavesdropping and Masquerading [6]. The design of privacy protecting mechanisms is a challenging problem for the intrinsic characteristic of WMSNs. There are two different types of solutions that aim at hindering such as attacks: The first of types is privacy aware mechanisms based on *data cloaking*. The aim of data cloaking anonymity mechanisms is hiding the informative content of messages by perturbing data according to specific patterns. There are only a few more prior studies on the issue of data cloaking , mainly considering the privacy such as [7] is expressed designed to enable privacy in vision rich system built in WMSN, [8] proposed a novel paradigm for securing privacy and confidently in a distributed manner, [9] presented attacks that affect the data privacy in visual sensor networks and proposed privacy-promoting security solutions established upon a detected-adversary using a game-theoretic analysis and keyless encryption. The second of types is privacy policy. The references of [10] propose privacy policy and they state who can use individuals data, which data can be collected, for what purpose the data can be used, and how they can be distributed. A privacy-preserving video surveillance system that monitor subjects in an observation region using video cameras along with localized sensors is presented in [11] . The localized sensors include RFID tags placed within the observation environment. The motion detectors are used to turn the video cameras on or off, while the RFIDs of the subjects provide information that specifies which individuals are entitled to privacy. The video data accommodates the information from the various sensors, and the result is in a video stream with only authorized subjects being masked through image processing. At present, the solutions that guarantee the privacy of data in the context of WMSNs are still in a primitive state and many open problems still exist such as lack of privacy and process data complements based on digital watermarking technique and are yet to be discovered, hence, further research work is required.

### 2.2 Authentication mechanisms

Wireless communications make security and privacy requirements critical take into account they increase the vulnerabilities and the threats on the integrity and confidentiality of the transmitted data. With these reasons,

there are many studies on the issue of an authentication mechanisms such as Honggang W et al [12] presented an authentication mechanisms that it is used to guarantee the correctness and the confidentiality of data and Zhang W et al [13] proposed an end-to-end, watermark statistical approach for data authentication that provides inherent support for in-network processing. Due to the high number of sensor nodes, such systems could contain control units that broadcast commands and data to the nodes. Consequence, the authenticity of these data and commands is a critical requirement for the correct behaviour of WMSNs. It is really a complex problem to guarantee the correct broadcast authentication of the messages transmitted by control units, because the broadcast authentication algorithms that are currently available in the literature [14, 15] do not adequately satisfy the QoS requirements of multimedia signals. At present there is no solution to deploy for WMSNs. Therefore, exploiting the characteristics of multimedia nodes should be developed.

### 2.3 Secure Communication Channel

The usage of secure communication protocols to hinder active attacks and eavesdropping is presented. In this case the cloaking is executed by means of encryption methods. The objective of these methods is to guarantee the confidentiality of data by hiding their content. There has been a lot work for securing routing protocol for WSN. A suite of security protocols for sensor networks called SPIN was recently proposed [16]. The SPIN family of protocols permits only valid key holders access to encrypted data; but as soon as this data is decrypted, tracking the reproduction or re-transmission of the data is not possible. The protocol SPIN originally designed for generic WSN can also be applied to WMSN. Like Fidaleo et al [17] introduce the Networked Sensor Tapestry (NeST) architecture which is designed for the secure sharing, capture, distributed processing, and archiving of multimedia data. The infrastructure of the NeST is developed to facilitate the fast prototyping and deployment of WMSNs for a wide variety of surveillance applications including structural monitoring and battlefield assistance. In order to facilitate trust in WMSNs, [5] presents the notion of subjective privacy in video where the behaviour of an individual under surveillance is conveyed but not identify it.

### 2.4. Secure Compression and aggregation of multimedia data contents

Aggregation algorithms and compression technique for multimedia contents are crucial to reduce the amount of transmitted data and to save energy and processing resources in WMSNs. The problem of aggregating multiple compressed frames coming from different video sensors while guaranteeing the expected security level is still open research area. Even though, many compression schemes have been proposed as described in the surveys [19]. Because of the complex compression operations, the distributed elaborate of the multimedia contents and the limited bandwidth and power resources of WMSN, so it is needed to introduce secure aggregation algorithms that decrease the total amount of information to transmit, elaborate and protect at the same time the quality of the multimedia message. There has been a lot of work in the

area of secure data aggregation such as Hani A et al [20] discussed the security issue in data aggregation in the WSN. A novel framework for secure information aggregation in large sensor networks is proposed by Bartosz P et al [21]. Wang et al [22] propose a survey on the most important solutions, but they can be hardly applied to multimedia data. In the case of WMSN, aggregation is probably, only going to be useful with abstract information extracted from sensed media. This is because it is extremely complex to aggregate different multimedia sources into a single aggregated multimedia stream [6]. To take into account the cost deriving from the secure aggregation of multimedia contents, [23] proposed a methods to optimize the placement of aggregation node. At present, research on secure aggregation of multimedia contents is still separated and that more efforts are required to address it in a comprehensive way.

### 2.5. Research issues for WMSNs Security

After doing the literature review, we identify the gaps in the following area.

1. Lack of privacy based on digital watermarking technique.
2. Lack of data authentication based on digital watermarking technique.
3. Lack of the processed data complements.
4. Lack of secure communication model based on watermarking technique for WMSNs.

In this context of this research, we will address and then issues. The next section will provide a conceptual framework to solve the open problem

## 3. CONCEPTUAL FRAMEWORK FOR SECURE COMMUNICATION IN WMSNs USING WATERMARKING

### 3.1. Overview Digital Watermarking and WSNs

The objective of digital watermarking is to protect the intellectual property of multimedia contents such as copy right protection, contents archiving, Meta-data insertion, broadcast monitoring, tamper detection and digital fingerprinting [18]. Digital watermarking techniques have been extensively studied in the multimedia domain [19, 20]. However, rarely this technique has been used in WSNs. Zhang W et al [13] propose a watermarking-based authentication schemes for WSNs. The key idea is to hide certain information about the multimedia material within that material itself. As illustrated in Fig. 1, a generic watermarking system is usually composed of two components: an embedder and a detector [13]. The embedder takes three inputs: (1) messages that are encoded as the watermark; (2) cover data that are used to embed the watermark; and (3) key

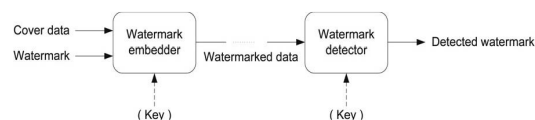


Fig. 1, A generic watermarking system [13]

That is optional for enforcing secure watermark generation. As an embedder's output, the watermarked data is distributed. It is presented as the detector's input, with the

key information (depending on whether employed), the detector can determine whether a watermark exists and decode it. The watermark detection schemes can be categorized into two classes: *informed detection* and *blind detection*. The difference lies in that the original cover data is accessible in the former case while it is not required in the latter case. Although watermarking technique has been widely used in the multimedia domain, its direct adoption to wireless sensor networks is often not feasible. First, often in multimedia applications, both the watermark embedder and detector (shown in Fig. 1) possess the knowledge of the whole multimedia material, which can be leveraged to watermark embedding. On the contrary, in sensor networks, each sensor node only has knowledge of its own local sensory data without a global view of the whole "sensory image". This requires the sensor node to embed its watermark in a distributed fashion [13]. Second, most watermarking schemes for multimedia applications operates in the frequency domain, for example, after certain time to frequency transform. However, in sensor networks, only the sink that performs the compression can obtain such information.

### 3.2. Research method

A science and engineering based research approach is adopted in this research project. Science and engineering research leads to the development of new techniques, architecture, methodologies, devices or a set of concepts, which can be combined together to form a new theoretical framework. This research approach commonly identifies problems and proposes solutions to these problems. [21] and [22] provide a concise conceptual framework for design-science research and state that design-science research deals with understanding the problem domain and design a solution by building application or some design artifacts.

### 3.3. Research Stages

The work in [1] gives a vision of the research challenges and the future trend focusing on their security aspects in WMSNs and the work [5] gives a driving directions for future research in secure WMSNs. In [5] shows that privacy, trust management and authentication mechanisms are not separated components of security in WMSNs.

Digital watermarking technique is an effective vehicle to assure and assert the image data authentication and is not inherently pose risk to privacy. At the same time, it relies on other security services. Digital watermarking technique and other security services together make up security architecture for WMSNs. Therefore, a comprehensive consideration is compulsory when designing digital watermarking for WMSN. Here we will propose a conceptual framework for watermarking enabled secure communication in WMSNs. It can be depicted in Figure 2. The conceptual will provide a guideline to design watermarking technique for WMSNs. The concept consists of 8 stages.

#### Stage 1 : Application scenarios extraction.

This step defines QoS requirement for different application scenarios extraction. WMSNs are application-specific networks. Except from some common features, a sensor network for a specific application has some features and the secure communication requirements. Suppose a multimedia

sensor network is deployed in the hospital surveillance environment [3] and the other in military [23]. Both network secure communication requirement should be different based on the resource of node can be used. The risks they face with. Therefore, we have to fully understand application background. The acquirement information in this step includes the size of the network and the densities, the available software and hardware resources, and some special knowledge that can be used in a real time, for example location information

**Stage 2: The secure communication model.**

This step develops the secure communication model. Since the protocol SPIN [16] originally designed for WSN and it can also be applied for WMSNs. So the secure communication model will be developed according to security requirements [16]. So far we know that it is the first time to define secure communication model with this terminology. Development of metrics, measurement and evaluation of approach are of great importance in order to establish a scientific methodology for the WMSNs area.

**Stage 3: Privacy protection for WMSNs.**

This step develops a privacy protection. WMSN collect a great amount of data, which may be used to violate individual's privacy, privacy protection is required. There are many existing privacy protection [24] but there is no the privacy of data in the context of WMSNs. [25] states that digital watermarking does not inherently pose risk to privacy and suggestions specific privacy principles for digital watermarking. One of the suggestions is privacy by design. Here, the privacy protection will be incorporated into the design of digital watermarking. The developing of digital watermarking considers and addresses privacy issues in the early design.

**Stage 4: Authentication for WMSNs:** This step defines an authentication mechanism. Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. Since data is easy to threat in wireless communication, so it is needed an authentication mechanism. The authentication mechanism rules privacy and security. [12] and [13] proposed authentication mechanism based on the digital watermarking technique. Here, we also addressed the digital watermarking technique as an authentication mechanism. This mechanism is used to guarantee the correctness and the confidentiality data for WMSNs.

**Stage 5: Trust management for WMSNs.** This step develops a trust management for WMSNs. The concept of trust is to increase security and reliability in sensor networks [26, 27]. We know that reputation is the opinion of one WSN node about another. The trust is a derivative of the reputation of an entity. The sensor network may be deployed in entrusted locations. We assume that individual sensors network is entrusted. Using SPIN [16], we compromise of a node to other nodes. Here the developing of digital watermarking considers and addresses trust management.

**Stage 6: Initializes system parameter.**

This step initializes system parameter. An authentication mechanism, used to guarantee the correctness and the confidentiality of data, is fixed in this step. To assert and

assure the data authentication digital watermarking technique is applied. Digital watermarking consists of two components: an embedder and a detector [13]. There are two inputs in the embedder. ie., messages that are encoded as the watermark, data comes from physical world. The watermark detector can determine whether a watermark exist and decode it. A fully integrated view of the design factors promotes the development of protocol for WMSNs

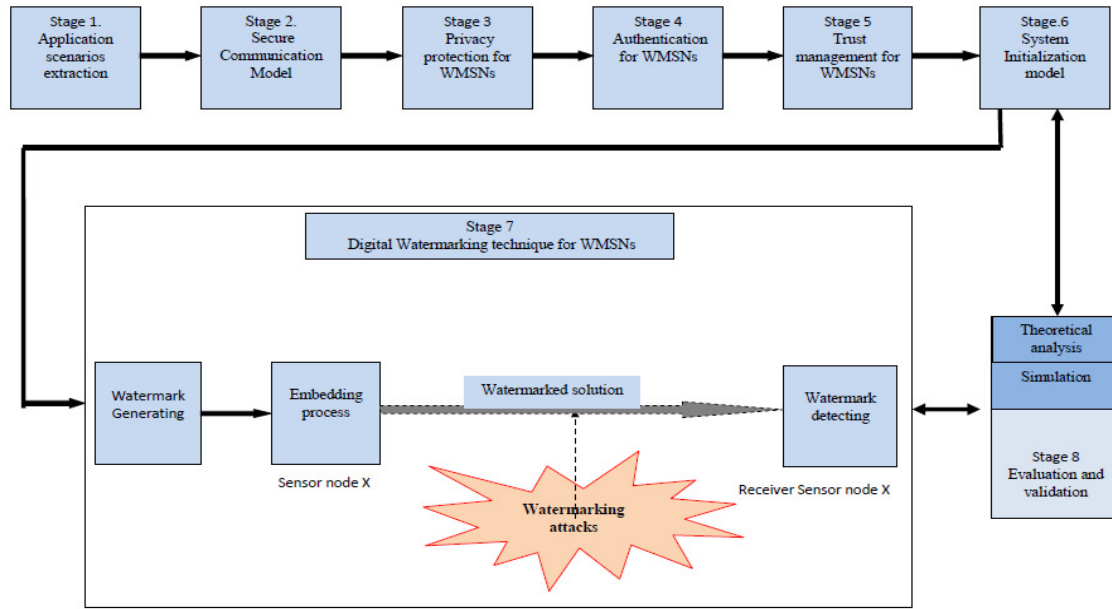
**Stage 7: Digital watermarking technique for WMSNs :**

This step explains how digital watermarking works for WMSNs. In this step, there are three stages. The first stage is generating watermark. Here, a watermark binary stream is generated by using Linear Feed Back Shift Register [20] , then it is converted to watermark constraint using a particular low of the kolmogorov rule [28]. The second stage is the process of embedding. Here, the watermark constraint and the original data which comes from physical world flow into the sensor node are processed into a format suitable for using multi-modal fusion. After that a non linear system equation is obtained using atomic acoustic trilateration,. Furthermore, the non linear system equation is solved by using the standard non linear programming approach to get watermarked solution. At last, the watermarked solution flows from sensor node to receiver sensor node. There are some watermark attacks which also flows from another sensor node into receiver sensor node. The third stage is the process of detecting watermark. Watermarked solution is detected in receiver sensor node whether it exists or not by applying the blind detection. This stage corresponds to perceptual level of the science and engineering research method.

**Stage 8: Theoretical analysis and simulation:** Both theoretical analysis and simulation are good tools to test the designed schemes. Only by these tools can we prove the validity of the schemes and, at the same time, find deficiencies of the scheme. The step 6, 7, and 8 proceed in turn. They form a loop and perform numerous times before the error minimization is obtained. This stage corresponds to the practical level of the science and engineering research methods

#### 4. FRAMEWORK IMPLEMENTATION

Based on the conceptual framework proposed in the first research stage, the main task in this stage would be to design secure communication protocols in WMSNs using digital watermarking. Each of these schemes has some specialties according to watermarking attacks. It is not the final aim and it is impossible to design one of type protocol which will outperform all other for all watermarking attacks. We devote to designing secure communication in WMSNs using digital watermarking which matches the abstracted model in Fig. 2. This design can be categorized into two paths: the process of embedding into sensor node, and the process of detecting from receiver sensor node. The process of embedding consists of two stages: firstly, generating a watermark binary by using Linear Feedback Shift Register (LFSR)[20]. Then converting the watermark stream into watermarking constraints using a particular low of the Kolmogorov complexity rule [28]. Secondly, the original data, gathered



**Figure 2 : framework for secure communication in WMSNs using watermarking**

from physical world, flows into the sensor node and then process, that data into a format suitable for multi-modal sensor fusion [29]. Here we use atomic acoustic trilateration [30]. In order to get non linear system equations those consists of objective FN, constraints and add constraints. To get a watermarked data, we have to solve the non linear system equation by using a gradient projection or the standard non linear programming approach. The process of detecting watermark uses *blind detection*. The blind detection means that we do not use the original data for watermarking detection. Here we approach by statistically analyzing the relationship between correctness, strength of authorship and measurement error [31].

### 5. VALIDATION

The schemes and protocol will be evaluated by using experimental simulation which can be divided into three stages: Stage 1, Here we will verify whether the schemes satisfy the secure communication data requirements against different watermarking attacks, such as Ghost signature, addition of a new signature, removal of the author's signature and de-Synchronization with the help of software Mathematica and Matlab. To this time there were limited references existed in experimental simulation, namely [30] and [32]. Both of them developed the system of watermarking techniques for embedding signatures into data and information acquired by embedded WSNs. However, [30] and [32] do not provide any attempt to handle some watermarking attacks. Thus, this research will contribute to the source studies on the field, providing digital watermarking schemes for WMSNs through experimental simulation. With regards to protocols, it is necessary to have

comprehensive guidelines for evaluating a specific protocol and compare it against others. Based on the proposed secure communication model, appropriate performance metrics would then be used to evaluate the strength and weakness of each protocol. Stage 2, The performance of the proposed secure communication schemes will be further simulated by NS-2. Stage 3, The security of the watermarking enabled secure communication schemes will be validated by mathematical security proof.

### 6. CONCLUSION AND FUTURE WORK

This paper aims to address the problem of secure communication in wireless multimedia sensor networks using digital watermarking. Although some work has been done in this area, there is no security in application on wireless multimedia sensor networks. The unique idea proposed in this paper aims to address the problem from a scientific and systematic process. Our future work is to provide watermarking enabled secure communication framework in WMSNs. This framework is focusing on establishing multimedia data authentication, and ensuring privacy perseverance in WMSNs.

### REFERENCES

- [1] Manel Guerrero Zapata, R.Z., Jos'e M. Barcel'Ordinas, Kemal Bicakci, Bulent Tavli, *The Future of Security in Wireless Multimedia Sensor Networks*. 2009.
- [2] Jason, C., Phillip, B. Gibbons, Suman, Nath, Padmanabhan, Pillai Srinivasan, Seshan Rahul, Sukthankar, *IrisNet: an internet-scale architecture for multimedia sensors*, in *Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, ACM: Hilton, Singapore.
- [3] Reeves, A.A., *Remote monitoring of patients suffering from early symptoms of dementia*. in Proc. Int. Workshop Wearable

- Implantable Body Sensor Networks: p. London, U.K., Apr. 2005.
- [4] Besma, R.A., Nash, R. Aragam, Yi, Yao Mongi, A. Abidi, *Survey and analysis of multimodal sensor planning and integration for wide area surveillance*. ACM Comput. Surv., 2008. **41**(1): p. 1-36.
- [5] Grieco, L.A., Boggia, G, Sicari, S, Colombo, P. *Secure Wireless Multimedia Sensor Networks: A Survey*. in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBICOMM '09. Third International Conference on*. 2009.
- [6] Marco, G.G., Schelle, Ashish, Jain Rick, Han Dirk, Grunwald, *Privacy-aware location sensor networks*, in *Proceedings of the 9th conference on Hot Topics in Operating Systems - Volume 9*. 2003, USENIX Association: Lihue, Hawaii.
- [7] Kundur, D., et al., *Security and Privacy for Distributed Multimedia Sensor Networks*. Proceedings of the IEEE, 2008. **96**(1): p. 112-130.
- [8] Czarlinska, A. and D. Kundur, *Reliable Event-Detection in Wireless Visual Sensor Networks Through Scalar Collaboration and Game-Theoretic Consideration*. Multimedia, IEEE Transactions on, 2008. **10**(5): p. 675-690.
- [9] Czarlinska, A., W. Huh, and D. Kundur. *On privacy and security in distributed visual sensor networks*. in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. 2008.
- [10] Sastry, D., Marco, Gruteser, Xuan, Liu, Paul, Moskowitz, Ronald, Perez, Moninder, Singh, Jung-Mu, Tang, *Framework for security and privacy in automotive telematics*, in *Proceedings of the 2nd international workshop on Mobile commerce*. 2002, ACM: Atlanta, Georgia, USA.
- [11] Jehan, W., et al., *Privacy protecting data collection in media spaces*, in *Proceedings of the 12th annual ACM international conference on Multimedia*. 2004, ACM: New York, NY, USA.
- [12] Honggang, W., et al. *Energy-Aware Adaptive Watermarking for Real-Time Image Delivery in Wireless Sensor Networks*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008.
- [13] Zhang, W., Liu, Yonghe, Das, Sajal K, De, Pradip, *Secure data aggregation in wireless sensor networks: A watermark based authentication supportive approach*. Pervasive and Mobile Computing, 2008. **4**(5): p. 658-680.
- [14] Liu, D. and P. Ning, *Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks*. 2002, North Carolina State University at Raleigh.
- [15] Liu, D. and P. Ning, *Multi-Level microTESLA: A Broadcast Authentication System for Distributed Sensor Networks*. 2003, North Carolina State University at Raleigh.
- [16] Adrian, P., et al., *SPINS: security protocols for sensor networks*. Wirel. Netw., 2002. **8**(5): p. 521-534.
- [17] Douglas, A.F., N. Hoang-Anh, and T. Mohan, *The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks*, in *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. 2004, ACM: New York, NY, USA.
- [18] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. Marcel Dekke, 2004.
- [19] Harjito, B., *Watermarking of Image Reconstruct by Using Information Dispersal Algorithm* Master Thesis, Computer Science Department at James Cook University of queensland Australia 1999.
- [20] Harjito, B., *Watermarking Technique based on Linear Feed Back Shift Register (LFSR)*, . Seminar Nasional Konferda ke -9 Himpunan Matematika Wilayah Jateng dan DIY di FMIPA UNS 2003.
- [21] T.M. Salvatore and F.S. Gerald, *Design and Natural Science Research on Information Technology*. Decision Support System. **Vol. 15**: p. 251-266, 1995.
- [22] Herner A, M.S., Park J, Ram S, *Design Science in Information System Research*. MIS Quarterly. **Vol 28**(1): p. 75-105, 2004.
- [23] Dan, L.W., K. D. Yu Hen, Hu Sayeed, A. M., *Detection, classification, and tracking of targets*. Signal Processing Magazine, IEEE, 2002. **19**(2): p. 17-29.
- [24] Qun, N., Alberto, Trombetta, Elisa, Bertino, Jorge, Lobo, *Privacy-aware role based access control*, in *Proceedings of the 12th ACM symposium on Access control models and technologies*. 2007, ACM: Sophia Antipolis, France.
- [25] Technology, C.f.D., *Privacy Principles for Digital Watermarking* Keeping the Internet Open, Innovative, and Free 1634 I St., NW, Suite 1100, Washington, DC 20006 • v. +1.202.637.9800. • f. +1.202.637.0968 • <http://www.cdt.org>, 2008. **1**(29 May).
- [26] Saurabh, G., K.B. Laura, and B.S. Mani, *Reputation-based framework for high integrity sensor networks*. ACM Trans. Sen. Netw., 2008. **4**(3): p. 1-37.
- [27] Avinash, S., T. Joshua, and W. Jie, *DRBTS: Distributed Reputation-based Beacon Trust System*, in *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*. 2006, IEEE Computer Society.
- [28] Li, P.V.M., *An Introduction to Kolmogorov Complexity and Its Applications*. Graduate Texts in Computer Science. Springer, New York, second edition 1997.
- [29] Richard, R.B. and S.S. Iyengar, *Multi-sensor fusion: fundamentals and applications with software*. 1998: Prentice-Hall, Inc. 488.
- [30] F. Koushanfar, M.P., *Watermarking Technique for Sensor Networks: Foundations and Applications*. Book chapter, in 'Security in Sensor Networks', Yang Xiao (ed.), 2007.
- [31] Hernandez, J.R. and F. Perez-Gonzalez, *Statistical analysis of watermarking schemes for copyright protection of images*. Proceedings of the IEEE, 1999. **87**(7): p. 1142-1166.
- [32] Fang Jessica, P.M., *Real-time watermarking techniques for sensor networks* ProceedingS-SPIE The international Society for Optical Engineering (5020): p. 391-402 2003.