Recovering and Restoring Tampered RFID Data using Steganographic Principles

Madan Mohan[2], Vidyasagar Potdar[1], Elizabeth Chang[1]
[1]Digital Ecosystems and Business Intelligence Institute, Curtin Business School, Curtin University of Technology
Perth, Western Australia
Vidyasagar.Potdar@cbs.curtin.edu.au, Elizabeth.Chang@cbs.curtin.edu.au
www.debii.curtin.edu.au www.rfidtamperdetection.com

[2]DigiBee Microsystems Private Limited, V Floor, No. 25, Dr. Radhakrishnan Salai, Chennai, India
madanmohan.manokar@gmail.com www.dgbmicro.com

***Abstract*- Security is one major issue with RFID technology. Mainstream research in RFID security addresses the following security properties i.e. anonymity, confidentiality and authenticity, however it does not cater for integrity. In this paper we consider the fourth security property i.e. *integrity*. We try to solve the issue of *data recovery* after RFID data has been tampered. To address this issue, we present a novel steganographic solution, which embeds a secret pattern in the serial number partition of the RFID tag. This secret pattern is the data that we assume would most likely be the candidate for tampering, for example the manufacturer's and products details stored on the RFID tag. The main motivation for an attacker to tamper this data would be economic benefits like low logistics cost, or quicker custom clearance, and this can only be achieved by changing product details or manufacturer details on the RFID tag. The novelty of this scheme lies in the fact that we have applied steganographic principles to RFID tags; in comparison, most of the existing steganographic solutions are limited to images, or audio, or video applications. We term this scheme *ResTamp* because it is *restores tampered* data. This paper provides a detailed theoretical foundation for the *ResTamp* algorithm.**

I. INTRODUCTION

A RFID tag is an electronic device that holds identification data. Typically, the RFID tag is attached to items and contains a serial number, which is used to uniquely identify them. RFID technology uses radio waves to automatically identify items which have RFID tags attached to it.

This technology was initially developed with the aim to manage and track items in supply chain and logistics, but nowadays it is used in many other areas e.g. medical applications, manufacturing, retail, livestock tracking and tracking exact timing in sports events. As pointed out by RFIDExchange "*RFID applications are limited only by imagination*" [22]. It can be used any where and every where if possible.

RFID technology is composed of three main components; *firstly,* a RFID tag, which contains the identification number, *secondly,* a RFID Reader, which activates the tag to broadcast its identification number and *finally*, a RFID Middleware, which integrates the information from the reader to the backend database systems [16, 17]. This is shown in Fig. 1.

However at present, the main issues with RFID technology privacy, security, cost, reliability, deployment, and scalability. Several proposals have been put forward to solve these issues. Within the security umbrella, there are several open issues. One such issue is *data tampering*. If the data stored on the RFID tag is tampered, then the tag becomes useless, because it cannot convey any information. This is a major security concern. An initial proposal for *tamper detection* was proposed by Potdar, Wu and Chang (2005). However this solution only addresses the issue partially, because the tampered data cannot be recovered, it can only be quarantined to prevent its entry to the backend databases and ERP (Enterprise Resource Planning) systems.

In this paper we extend this work, by presenting a solution to recover tampered data from a collection of RFID tags, after data tampering has been confirmed. The proposed solution is based on the concepts derived from *information hiding* and *steganography*.

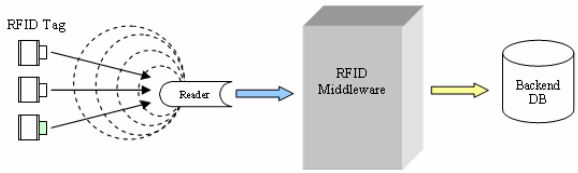

Fig. 1 RFID Architecture

The paper is organized as follows. In Section 2, we survey the existing literature on RFID security. In Section 3, we formalize the problem description. In Section 4, we propose the *data recovery* solution. In Section 5, we provide a discussion and conclude the paper in Section 6.

II. LITERATURE REVIEW

While researchers are just starting to address security questions, privacy advocates and legislators have for some time been attempting to address the privacy issues. A lot of work has been done to address the privacy issues in RFID deployment; however literature addressing the security issues is quite limited. The main aim of this section is to discuss the

security issues in RFID systems and survey the relevant literature that is proposed to address the same.

Wong and Raphael (2006) classify attacks on RFID systems into two categories – passive attacks and active attacks. *Passive attackers* are those who eavesdrop on the communications channel, but do not affect or interfere with the communication in any way [27]. Passive attacks compromise the *confidentiality* and *anonymity* in communication. Consider the warehouse management scenario, if a malicious reader can eavesdrop (spy) the communication between the tags and the readers, *confidentiality* and *anonymity* in such communication is lost because the entity involved in the communication is unaware when it is being attacked.

Active attackers are those who directly interfere with the communication of messages, either by interrupting, fabricating or modifying communicated messages [27]. Active attacks compromise the *availability*, *authenticity* and *integrity* in communication. *Interruptions* refer to denial of service attacks

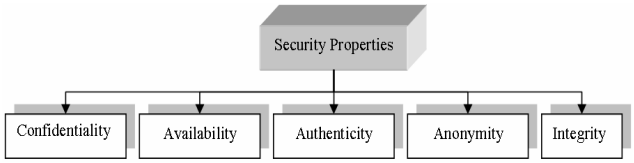

Fig. 2 Major Security Issues with RFID Adoption

(*availability*) on RFID tags. Engberg, Harning, and Jensen (2004), argued that if such an attack is launched, the RFID reader cannot query the tags which could have an adverse effect on a warehouse management system (or related applications) as it may stop responding and real-time status of the warehouse cannot be made available [4].

Fabrication refers to attacks on the *authenticity* of the information on the RFID tag, such as tag forgery. RFID tags might be forged in order to get access to restricted locations within an organization.

Modifications refer to attacks on the *integrity* of the information on the RFID tag (or system) such as data tampering. Data on the RFID tag could also be tampered with by malicious readers. Consider the warehouse scenario once again: if the data on the tag is tampered with, it could result in shipping wrong items from the warehouse. For instance, if the malicious reader changes the information on RFID tag from Orange to Apple, then a palette containing Apples might be shipped when the intention was to ship Oranges. Data tampering (or integrity) can raise issues like QoS (Quality of Service) and Trust in logistics and supply chain and hence needs to be addressed thoroughly.

We now discuss the current literature which addresses some of the security issues highlighted in the Fig. 2. Most of the proposed solutions discussed here address the first four security properties i.e. confidentiality, availability, authenticity and anonymity. We begin the discussion with solutions to manage anonymity.

A. Anonymity

RFID technology shows the characteristics that can invade personal privacy; hence anonymity is highly desired if this technology would be deployed in mass scale. A lot of work has already been conducted in this area and several proposals are put forward to address the issue of privacy [6, 8, 9, 11, 12, 13, 14, 17, 23, 24, 25, 26, 27]. In this section, we discuss several approaches that can be used to provide consumer privacy.

One of the simplest approaches to address the issue of privacy is to kill the tag once it leaves the supply chain and enters the consumer market. This approach is used by EPC standard, which make the tags permanently inoperative. It is envisioned that the point-of-sale (POS) operator would have RFID reader that can send the command to kill the tag once it is sold to the consumer. However, to address the issue of malicious tag writes, the kill command is protected by a secret PIN, which in this case is assumed to be with the POS RFID reader. Another approach is to add a RFID tag on the price tag. Hence, when the price tag is removed, the RFID is removed as well and can guarantee privacy. However, as pointed out by Juels (2005), removing or killing the tags can restrict the post purchase benefits of RFID tags like receiptless item returns [11]. As a result, it would be useful if the tags could be temporarily deactivated. This could be achieved by access control mechanisms similar like using a PIN. Several other approaches to anonymity and privacy are outlined in Table 1.

TABLE 1
ANONYMITY (PRIVACY)

Proposal	Approach
Inoue & Yasuura 2003	Using two tags – one for unique identification and other for product details. Does not address clandestine inventorying or tracking.
Juels and Pappu 2003	Re-encrypting the tag content using El-Gamal cryptosystem. The solution is presented in the context of securing RFID enabled banknotes.
Juels, Rivest & Szydlo 2003	Blocker Tags: A tag that specifies whether it can be read or not. A privacy bit (0 or 1) is assigned on the tag, which determines whether the tag can be publicly scanned (bit 0) or can be used privately (bit 1).
Ateniese, Camenisch & de Medeiros 2005	Proposed to use bilinear pairing in elliptic curve cryptography. Authenticity of the tag identifier is maintained by digitally signing the ciphertext with a trusted CA. This approach cannot address the issue of ciphertext swapping, i.e. when eavesdropper changes the content of two RFID tags simultaneously by swapping their content.
Rakesh Kumar	A Faraday cage is an enclosure designed to exclude electromagnetic fields. As a result, certain radio frequencies cannot penetrate through it. It can address privacy concerns, e.g. if high values currency notes start embedding a RFID tag, then using foil lined wallets can guarantee privacy

B. Confidentiality

Several approaches to access control are proposed in the literature. We will discuss a few of the approaches in greater detail in this section.

Juels, Rivest and Szydlo (2003), discuss a hash based Access Control Protocol [13]. Here the tag is first in a *locked state*. When the tag moves to the *unlocked state* the reader can access the tags details. In order to change the state the tag first

transmits Meta ID' which is the hash value of a key. An authorized reader looks up the corresponding key in a backend system and sends it to the tag. The tag verifies the key by hashing it, returns the clear text ID, and remains only for a short time in an 'unlocked' state which provides time for reader authentication and offers a modest level of access security.

C. Authenticity

The literature on the authentication of the RFID tag is also very mature as of today. Several proposals are presented in the domain of tag authentication, reader authentication, and anti-counterfeit tag. Some of these approaches are outlined in Table 2.

TABLE 2
AUTHENTICITY

Proposal	Approach
Juels 2005	PIN: Authenticate the tag to the reader
Juels 2004	Yoking Proofs – provides cryptographic proofs that two tags were scanned simultaneously and in physical proximity. Can be used in a pharmacy to prove to a government agency that the pharmacy scanned a RFID tagged medicine bottle and delivered the exact medicine as prescribed on the RFID tagged prescription
Engberg et al. (2004)	Zero-knowledge based protocols for communication between reader and tag so that they can authenticate each other without revealing any secrets that may allow them to be tracked.
Molnar & Wagner, 2004	Mutual authentication schemes using challenge-response based on the use of pseudo-random function in the computation of responses to challenges.
Feldhofer et al., 2004	Proposes the Simple Authentication and Security Layer (SASL) protocol with AES encryption and analyses the hardware requirements
Dimitriou, 2005	Provides forward secrecy by using nonces (random numbers that are never reused) by both the reader and tag in their challenges to each other.

This concludes the survey of the most relevant literature on RFID security. We observed that most of the solutions addressed the issue of authentication, confidentiality and anonymity. Existing solutions do not address the issue of data integrity of the RFID tag in detail.

III. PROBLEM DESCRIPTION

Data tampering and data recovery are the two major security concerns in RFID deployment. Any solution that can address these issues i.e. to detect that data tampering has happened and to provide a mechanism to recover the tampered data would be very significant, whenever large and complex RFID based solutions require security and authenticity.

One good example would be *e-logistics* and *e-Warehouse* where *consortia* of small to medium enterprises (SME) around the world work together to share businesses, customers, resources and goods tracking to provide just-in-time customer services. Such a *virtual collaborative environment* survives on the assumption of *reliable* and *authentic* information. Automatic identification technology provided by RFID heavily relies on the authenticity of the information. If this information is tampered, it can *destroy* the *reputation* of the *businesses* collaborating in the virtual environment. In distributed logistics networks and extended enterprises, collaborating peers could accuse each other for being vulnerable to security attacks which may *reduce* their *trustworthiness*, and eventually such a collaborative environment would not sustain any longer. For example, if the data on the RFID tag representing the '*nature of good*' were changed from '*Mangoes'* to '*Oranges'* wrong goods would be shipped to wrong customers, which would in turn *affect* the *reputation* of the logistics provider. Such acts could be organized by competitive organization in an attempt to thwart the reputation of logistics providers.

This demonstrates the need for solutions, which can offer *tamper detection* and *data recovery* after tampering has been identified. Initial work on tamper detection was presented by the authors earlier [19, 20, 21, 28]. However after conducting a detailed literature survey of RFID security solutions we identified that no one has yet presented a solution to address the issue of data recovery after data tampering. This gives us the rationale to present our solution for data recovery after data tampering, which is based on the principles of information hiding and steganography.

IV. PROPOSED DATA RECOVERY SCHEME – RESTAMP

In this section, we give a general overview of *ResTamp* (pronounced as *Re-stamp*) solution, followed by eliciting the main requirements for *ResTamp*. Based on these requirements the design rationale for *ResTamp* is outlined where we discuss the basic design decisions.

A. General Overview of the ResTamp Solution

ResTamp offers a steganographic solution to recover tampered RFID data. The proposed solution relies on [19, 21], to ascertain that data tampering has occurred and to identify which portion of the RFID tag that has been tampered with. In the proposed *ResTamp* solution, we assume that only the EPC Manager (EM) and the Object Class (OC) would be tampered. This is because we assume the intentional attack or the tampering attempt, has been driven by economic motives, like reduced transportation cost or easier entry at overseas ports. And the only way to achieve this is by modifying the EM or the OC component of the RFID *data structure*[1]. For example, OC is used to uniquely identify one product; if product A (Orange) has cheaper transportation cost compared to product B (Mango), the attacker might attempt to change OC of product B, to gain an economic benefit. However if the attacker changes the serial number (SN), which is used to identify one item of a specific product, it cannot gain any economic benefit, because SN does not represent a product, but just a unique identifier for a item belonging to one product.

[1] RFID data structure is composed of four partitions – Header, EPC Manager (EM), Object Class (OC) and Serial Number (SN). EM uniquely identifies a manufacturer globally, OC identifies one product manufactured by one manufacturer i.e. EM and SN identifies one unique item belonging to one product. A detailed explanation can be found here [19,20,21].

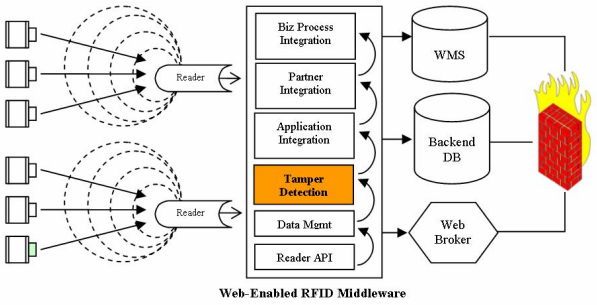

Fig. 3 RFID Middleware Architecture with Tamper Detection Facility

To achieve data recovery, *ResTamp* embeds a *secret pattern,* generated by using EM and OC, within the SN partition of all RFID tags belonging to one product or a consignment. This embedded pattern is used by the data recovery module to generate the tampered data. This is done in the following manner: suppose the secret pattern is *n* bits long, we embed *m* bits in each RFID tag. Hence a total of [*n*/*m*] RFID tags would be used to hide the secret pattern generated using EM and OC. If we assume *m* = *1*, then the first bit is embedded in the RFID tag which has the lowest serial number. As shown in Table 3[2], the RFID tag with the serial number '*1111112345*' would be used to hide the first bit of the secret pattern. In the similar manner all the other bits in the secret pattern would be added to the remaining RFID tags. If the secret pattern was represented by 7 bits as '1010101', the corresponding bits in the RFID tag would be changes as shown in Table 3. In Table 3, we used the 3rd most significant bit (MSB) of the SN to hide the secret pattern. All the numbers with a *bold* and *underlined* facing when combined together would represent the secret pattern.

TABLE 3
SAMPLE RFID SERIAL NUMBERS

Serial Number	Decimal SN
10**1**01010101010101110101010101010000	1111112345
10**0**01010101010101110101010101010001	1111112346
10**1**01010101010101110101010101010010	1111112347
10**0**01010101010101110101010101010011	1111112348
10**1**01010101010101110101010101010100	1111112349
10**0**01010101010101110101010101010101	1111112350
10**1**01010101010101110101010101010111	1111112351
10**0**01010101010101110101010101010001	1111112352

When the EM or OC is tampered, we can recover the original values by resorting to the embedded pattern in the SN partition. The detailed algorithm is explained later.

The functionality for embedding of the secret pattern is assumed to be present in the RFID reader which initially writes the tags, whereas the extraction algorithm is assumed to be available as a component which can be plugged in the RFID middleware applications. The *ResTamp* solution would be a part of the tamper detection component in the RFID middleware and is shown in Fig. 3.

This component takes input data from the data management layer, and then detects whether EM or OC is tampered. If it is tampered then the secret pattern is extracted using all the RFID tags used for embedding the secret. The tampered data can then be restored and then propagated to the application integration levels in the middleware architecture.

B. *Requirements for ResTamp Solution*

In order to address the issues of data tampering, the following requirements are laid for the proposed TamDetect solution.

1. *Length of Secret Pattern*: The length of the secret pattern should be less than or equal to the total number of bits available in one set of RFID tags, which can be used for embedding. It should not occupy a lot of space because the amount of data that can be stored on a tag is very limited.
2. *Secret Pattern Generation*: The inputs for generating the watermark should be available on the tag itself.
3. *Embedding Locations*: The secret pattern should be embedded in the serial number partition.
4. *Data Recovery*: The algorithm should be able to recover tampered data after data tampering has been confirmed using TamDetect.
5. *Plug-n-Play Architecture*: The proposed solution should be designed such that it can be easily plugged into existing RFID middleware applications.

C. *Design Rationale for ResTamp Solution*

The theoretical foundation for *ResTamp* is proposed to satisfy the requirements outlined above. The following design decisions are proposed in this solution.

1. The size of the secret pattern is limited to *fifty two bits* and additional error correction bits (*e*). Assuming each bit is embedded in the one RFID tag; ResTamp would at least require [*52*+*e*] RFID tags. The 52bits represent the 28 bits from EM, 24 bits from OC. (Req. 1)
2. The secret pattern is generated from the data stored in the EM and OC. (Req. 2)
3. The secret pattern is embedded in the *serial number partition* because it offers enough bits (36 bits), which can be used for embedding. (Req. 3)
4. The secret pattern would be extracted from the set of RFID tags, to recover the tampered data. (Req. 4)
5. The algorithm is designed as a component; hence, it can be easily plugged into any existing middleware application. (Req. 5)

We now discuss the theoretical foundation for ResTamp.

V. THEORETICAL FOUNDATION FOR PROPOSED DATA RECOVERY SCHEME – RESTAMP

The proposed framework is shown in Fig. 4, 5 and 6. It can be decomposed in four different stages:

1. Secret Pattern Generation

[2] Example of 8 RFID tags and their serial numbers are as shown in Table 3. Consider the SN in decimal for easier understanding, although in reality these are represented using binary or hexadecimal.

2. Selecting the Embedding Location
3. Secret Pattern Embedding
4. Secret Pattern Extraction for Data Recovery

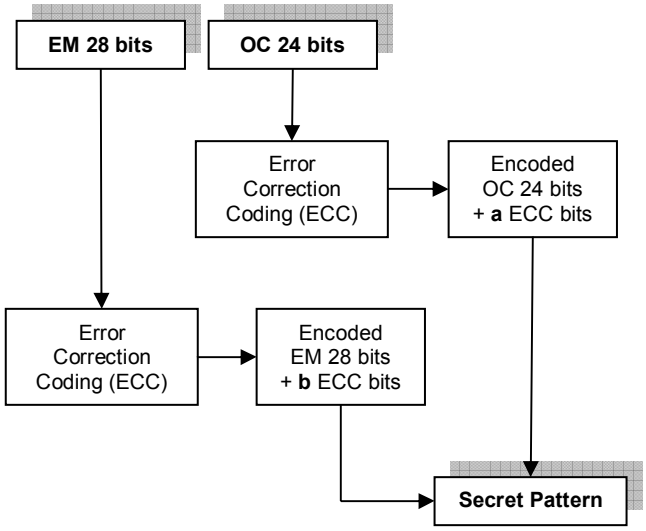

Fig. 4 Secret Pattern Generation

A. Secret Pattern Generation

Inputs	*EPC Manager (EM), or Object Class (OC)*
Outputs	*Secret Pattern (S_f)*

Step 1: Secret Pattern Generation

The secret pattern is generated as shown in Fig. 4. One instance of EM and OC, which is uniform across a given product category, is encoded using *error correction codes* (ECC) to offer additional level of security and also helps in recovery of tampered data. EM introduces an extra '*b*' bits along with 28bits and OC introduces another '*a*' bits along with 24bits. Error control encoding helps in recovering lost information in communication systems. The ECC is mostly useful in case the SN in some RFID tags is tampered. In that case, the resulting secret pattern would be incorrect and hence ECC would be required to regenerate the lost information from a completely tampered RFID tag. Once the secret pattern is generated, we have to identify the location for embedding. We now discuss how we select the appropriate location for embedding the secret pattern.

B. Selecting the Embedding Location

Previously, we mentioned that the secret pattern (S_f) should be embedded in the serial number partition of the RFID tag. In this section, we give the reason for this selection.

The basic principle of steganography (or information hiding) is that we need some redundant space within the host signal which can be modified to embed the secret pattern. In this case, the RFID tag is the host signal and we want to identify the redundant space. In order to do this, we investigated the RFID data structure.

On the basis of that investigation, we determined that the serial number partition within the RFID tags can offer a reasonable amount of redundant space for embedding the fragile watermark. This selection is attributed to the following facts:

The *Header*, is fully used for identifying the EAN.UCC key and the partitioning scheme. Hence, there is no redundant space, so there is no possibility for embedding the fragile watermark.

The *EPC Manager*, is used to identify the manufacturer uniquely. Hence, this partition also does not offer any redundant space for embedding because it might be decided by the industry standard and the manufacturer has least control over this.

The *Object Class*, is used to identify the product manufactured by the manufacturer. It may follow some product convention taxonomy where the *first* two digits might represent the classification of that product; the next two may be the age of product and so on. Hence, modifying any of this data might interfere with the existing industry standard. As a result, this partition also does not offer enough room for embedding the watermark.

The *Serial Number*, which is the last partition, is used to uniquely identify an item, which belongs to a particular Object Class. It is orthogonal to the *first* three partitions and can be decided by the manufacturer at will, without violating any existing industry standards. Consequently, it offers enough redundant space to embed the secret pattern. Meanwhile, the length of this partition is 36 bits (in EPC96) which offers enough room to accommodate the secret pattern. Thus, this becomes the most appropriate candidate for embedding the watermark, and hence, we decided to choose this partition for embedding. We now discuss the embedding and extraction algorithm in detail.

C. Secret Pattern Embedding

Inputs	*Serial Number (SN)* *Secret Pattern (S_f)* *Number of RFID Tags (N)* *Length of Secret Pattern (n)* *Number of bits embedded in one tag (M)* *Embedding Locations (L)* *Parity Bit Location (P)*
Outputs	*Tamper Proof RFID Tag (W)*

Step 1: Load the Secret Pattern

In the *first* step, the RFID reader loads the secret pattern in its memory. The secret pattern S_f is can be generated by the RFID reader or by the RFID middleware. We assume that the reader has the functionality to generate this.

Step 2: Select the embedding location within the serial number partition

The SN partition has 36 bits; we select $m+1$ consecutive bits from the SN partition, where ($0<m<36$), to embed the first M bits of the secret pattern. We express this location in the SN as L. The extra bit is used as a parity bit to check whether the secret pattern has changed after embedding.

Step 3: Append Parity Bit

In this step an even parity bit is appended in the next RFID tag i.e. N+1. For example, if the secret pattern is 3 bits in length (101) then we append a 0 to make it 1010. However the last bit i.e. '0', is embedded in the $(N+1)^{th}$ RFID tag. This provides additional security.
Step 4: Embedding the first set of secret pattern The *m* bits of the secret pattern are now embedded in the SN partition of the RFID tag. The process of embedding the secret pattern is shown in Fig. 5.

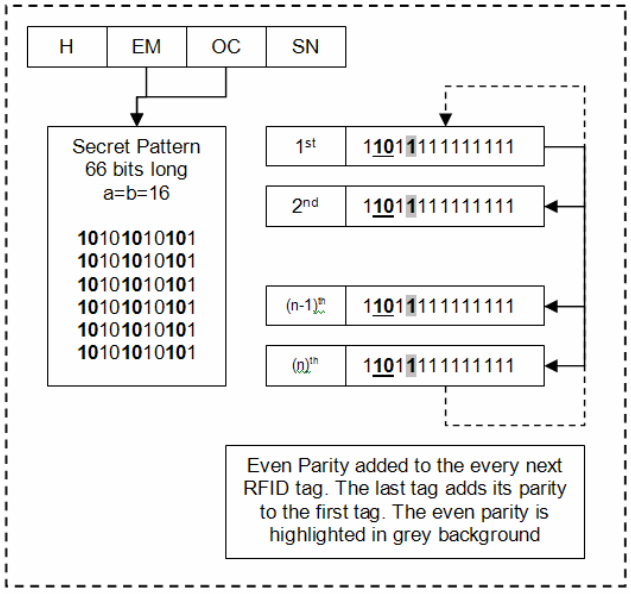

Fig. 5 Secret Pattern Embedding Algorithm

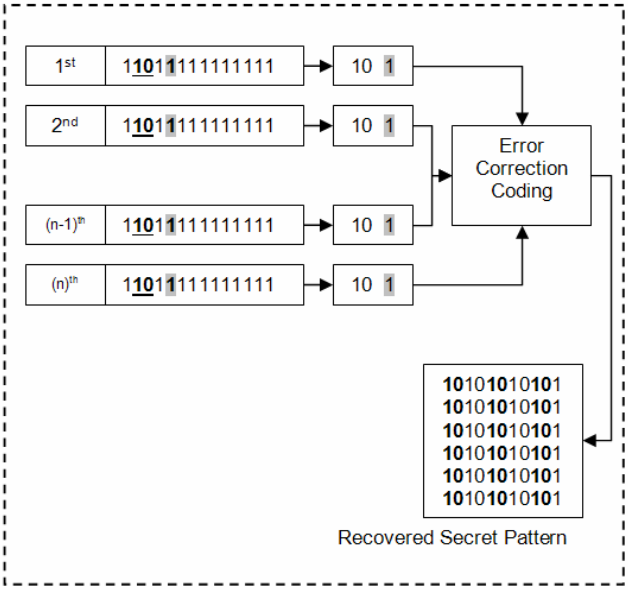

Fig. 6 Secret Pattern Extraction Algorithm

Fig. 5 shows the process of embedding the secret pattern in multiple RFID tags. Here we assume that the length of the secret pattern is 66 bits. 52 (28+24) bits are inputs to ECC, which results in 66 ((28+a) + (14+b)) bits output, where a + b =14 bits. The ECC output 66 bits (14 bits more than the input EM and OC) helps in recovering tampered bits in SN. The sum (a + b) can be increased to increase the number of tampered bits that could be recovered. We also consider the total number of RFID tags to be 33 in this case, hence we embed two secret bits in each RFID tag. The parity bit of the first RFID tag is embedded in the second RFID tag and so on. Similarly the parity bit of the last RFID tag is embedded in the first RFID tag. This is how the secret pattern is embedded in the tag. We now explain the data recovery algorithm.

D. Secret Pattern Extraction for Data Recovery

Inputs	*Serial Number (SN)* *Embedding Location (L)* *Parity Locations (P)*
Outputs	*Tampered Data Recovered*
Step 1: Check for SN tampering The first task is to check whether the secret pattern embedded in the SN is tampered or not. To check this we first look for even parity. → IF even parity exists in all the tags, we conclude that the SN partition in the RFID tag has not been tampered. → Proceed to the next step → ELSE IF the parity does not exist in one of the RFID tags we rely on Error Correction Coding. Error Correction Coding would fix these errors to a certain extent, depending upon how many bits are used for ECC. → Proceed to the next step	
Step 2: Generate the Secret Pattern Based on the extracted bits, recover the secret bits using ECC coding.	
Step 3: Provide the recovered data In this step, compare the extracted secret bits with the EM and OC, since data tampering has already been detected, rewrite the RFID tags with the recovered data.	

The process of data recovery from tampered RFID tag is shown in Fig. 6.

VI. Discussion and Validation

In this paper, we proposed a solution to recover data from tampered RFID tags. This was achieved by embedding a secret pattern in a group of RFID tags, which were attached to a consignment carrying one group of products. We showed how we can generate the secret pattern using EM and OC and embed that in SN. The proposed solution has many advantages; it is not only used for data recovery, but can also be used for secret communication. For example, this approach can be used to embed some other kind of information like *invoice data* or any other information that has to be shared between two communicating parties exchanging a consignment.

As long as the serial number has not been tampered with, the proposed technique can exactly recover the tampered data, which in this case is the EM and OC. But if the SN is tampered, to match the SN of another product (with lower transport cost), we recommend that the manufacturer should

follow a standard whereby the first *n* bits from the SN would be uniform across all its product range. These *n* bits would then be used for data recovery using information hiding or steganography. Thus we can recover data in case the SN is also tampered.

VII. Conclusion And Future Work

In this paper, we proposed a solution to recover data from tampered RFID tag. We found the majority of recent research work in RFID security has been done in the areas of anonymity, confidentiality and authenticity. Data integrity and data recovery has not been tackled in detail. Hence, we proposed a data recovery framework by introducing a layer into existing RFID middleware architecture. We also provided a detailed description of the data recovery algorithm, which can recover the tampered RFID data i.e. EM and OC.

References

[1] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. November 2005.

[2] Caspian: "Scandal: Wal-Mart, P&G involved in secret RFID testing,"Nov 10, 2003

[3] Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks, in Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05).

[4] Stephan Engberg, Morten Harning, and Christian Damsgaard Jensen. Zero knowledge device authentication: Privacy & security enhanced RFID preserving business value and consumer convenience. October 2004.

[5] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. 3156:357–370, August 2004.

[6] Lukas Grunwald, "RFDump Can Hack RFID Tags" , Available online: http://www.rfidgazette.org/2004/07/lukas_grunwalds.html Accessed on Sunday, 29 October 2006

[7] Dirk Henrici and Paul M¨uller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. pages 149–153, March 2004.

[8] G. V. Hulme, T. Claburn, "RFID's Security Challenge- Security and its high cost appears to be the next hurdle in the widespread adoption of RFID", in InformationWeek, Nov. 15, 2004 URL: http://www.informationweek.com/story/showArticle.jhtml?articleID=52601030

[9] Sozo Inoue and Hiroto Yasuura. RFID privacy using user-controllable uniqueness. November 2003.

[10] Ari Juels. "yoking-proofs" for RFID tags. pages 138–143, March 2004.

[11] Ari Juels. RFID security and privacy: A research survey. Manuscript, September 2005.

[12] Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. 2742:103–121, January 2003.

[13] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. 8th ACM Conference on Computer and Communications Security, pages 103–111, 2003.

[14] Heiko Knospe and Hartmut Pohl. RFID security. Information Security Technical Report, 9(4):39–50, November–December 2004.

[15] Rakesh Kumar. Interaction of RFID technology and public policy, Wipro White Paper, November 2003.

[16] Hennig, J. E., Ladkin, P. B., Siker, B., "Privacy Enhancing Technology Concepts for RFID Technology Scrutinized" 2005

[17] David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. pages 210–219, October 2004.

[18] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures" In Conference on Computer and Communications Security – CCS, ACM Press, 2004 pp. 210-219

[19] Vidyasagar Potdar, Chen Wu, Elizabeth Chang, "Tamper Detection for ubiquitous RFID enabled Supply Chain," In Y. Hao et al. (Eds.): CIS 2005, Part II, LNAI 3802 Springer-Verlag Berlin Heidelberg 2005 and Proceedings of the International Conference on Computational Intelligence and Security (CIS05), pages 273-278, 2005.

[20] Vidyasagar Potdar, Chen Wu, Elizabeth Chang, E-Supply Chain Technologies and Management, chapter "Automated Data Capture Technologies – RFID". IDEA Group Reference, Hershey, PA, USA. To Appear in March 2007, Accepted July 2006.

[21] Vidyasagar Potdar, Elizabeth Chang, "Tamper Detection in RFID tags using Fragile Watermarking," To Appear in the Proceedings of the IEEE International Conference on Industrial Technology (ICIT06), Mumbai, INDIA, 15-17 Dec, 2006.

[22] RFIDExchange http://www.rfidexchange.com/applications.aspx

[23] Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. 3207:879– 890, August 2004.

[24] William Stallings. Cryptography and Network Security. Prentice-Hall, Inc., 1999.

[25] Stephen Weis. Security and privacy in radio-frequency identification devices. Master thesis, Massachusetts Institute of Technology (MIT), Massachusetts, USA, May 2003.

[26] S. A. Weis, S. E. Sarma, R. L. Rivest, D. W. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems", in D. Hutter et al. Edn. Security in Pervasive Computing 2003, LNCS 2802, pp. 201-212, 2004

[27] Dennis M.-L. Wong and Raphael C.-W. Phan. RFID systems: Applications versus security & privacy implications, to be published by IDEA group, 2006.

[28] Manohar Potdar, Elizabeth Chang, Vidyasagar Potdar "Applications of RFID in Pharmaceutical Industry," To Appear in the Proceedings of the IEEE International Conference on Industrial Technology (ICIT06), Mumbai, INDIA, 15-17 Dec, 2006.