E-Forensics Steganography System for Secret Information Retrieval

Vidyasagar M. Potdar[1], Muhammad A.Khan[1], Elizabeth Chang[1]
Mihaela Ulieru[2] and Paul R. Worthington[2]

[1]School of Information Systems, Curtin University of Technology, Perth, Western Australia,
[2]Emergent Information Systems Laboratory, The University of Calgary, CANADA
e-mail: PotdarV, KhanM, ChangE@cbs.curtin.edu.au

Abstract — Steganography is the art and science of hiding information. This paper introduces e-Forensics as a novel technique for extracting secret information electronically encoded in most creative ways. We propose an e-Forensics system capable to detect (or extract) secret information using any generic steganalytic algorithm. The system is based on agent computing approach where the autonomic agent would traverse across several websites and detect any steganographic communication. If any such activity is detected it would report back to the concerned authority. Conceptual overview of the system, as well its design layout is presented. An illustrative example clarifies system's functionality and performance.

Keywords—Autonomic Agent, Steganography, Secret Information Hiding, Data Embedding, Stego-key, Data Extraction.

1. Introduction

Forensics is the investigative activity relating to the application of systematized knowledge resulting from observation, study, and experimentation carried on in order to determine evidentiary value of items [31]. According to Walter (1999), the word forensics is derived from the Latin forensis, which means belonging to the forum, where law courts were held in Rome.

As stated by Healy (2004) [32] e-Forensics is "the application of the information sciences to proprietary electronic devices such as cellular phones". McKemmish (1999) [33] further states that e- Forensics is the "process of identifying, preserving, analyzing and presenting electronic evidence in a manner that is legally acceptable". In our opinion e-Forensics has the express purpose of identifying, preserving, analyzing hidden, encrypted, deleted and destroyed (electronic) information. To protect secrecy, anti-social organizations continuously discover new and better cover mediums as well as design and develop robust algorithms.

Steganography which is also known as 'covered writing' includes methods of transmitting secret messages through innocuous cover mediums in such a manner that the existence of the embedded messages is undetectable. Using steganographic techniques we can hide secret information in digital image files, digital audio and video files, or any other digital media which has some redundant bits that can be replaced to hide secret data. Pre-computing steganography has a long history but digital steganography as a research field is avante garde.

We focus on the use of steganography to achieve the goal of disguising the existence of secret communication and steganalysis to recover secret communication. If the existence of communication is deciphered then this goal is

defeated, as such the e-Forensics system proves successful. We think steganalysis can be useful companion to e-Forensics.

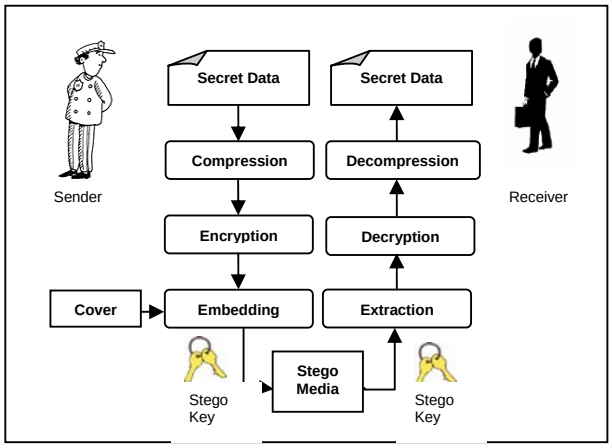


Figure 1: Generic Steganographic Technique

A generic steganographic technique is described in Fig. 1. The *Sender* wants to communicate a *secret message* to a receiver. The message is first compressed and then *encrypted*. The encrypted message can now be secretly hidden in a *cover medium*. A cover medium, as mentioned earlier, can be an image or any digital medium that has sufficient amount of redundant bits, which can be replaced to hide a secret message. A *stego-key* is generated and shared between the sender and the receiver. This stego-key is used to randomly select and replace the *redundant bits* from the cover media in order to hide the secret message. A stego-key can even be a cover media that is shared between the receiver and the sender. Redundant bits are defined as those bits in the cover media, which if changed won't change the cover media to a great extent. The *embedding* process hides the secret message using the stego-key. After embedding is finished the cover media can be transmitted to the receiver. At the receiving end, the *receiver*, having the proper stego-key and decryption key, can *extract* the secret message from cover media. The success of steganography is dependent on the secrecy of the cover media. Once the cover media is public then the success depends on the robustness of the algorithm used. We use steganography to achieve the goal of disguising the existence of secret communication.

2. Addressed Problem

In this paper we address the issue of secret communication by proposing e-Forensics Steganography System. Steganography which refers to secret and covert communication can be used by anti-social organizations to exchange secret messages innocuously. To detect (and possible recover) such messages form the cover objects (e.g. images) requires steganalytic techniques. Most of the current steganalytic techniques that have been proposed so far are not real time. Everyone assumes that steganographic cover medium to be available offline to be processed. In this paper we propose a real time e-Forensics System which would do real time steganalysis based on agent computing paradigm. We propose to use an agent based approach to identify any possible secret communication that may be happening. In this paper we give a basic structure of the proposed framework and explain it functionality with an example of any generic steganalytic algorithm. but before discussing the details of the proposal we would like to give some background of image steganography.

3. Image Steganography

Simmons first introduced the concept of steganography in the early 1980s when he discussed the prisoners' problem [24]. He discussed the situation in which two prisoners who are locked in different cells have to communicate innocuously without raising any suspicion. He used the idea of subliminal channels instead of steganography. This was one of the first works in the field of steganography.

There are a variety of techniques using which data can be *embedded* in images [2, 3, 4, 6, 12, 15, 17, 18, 20, 21, 22, 23, 25]. At the same time literature shows the existence of several techniques using which hidden data can be *detected* [1, 7-11, 14, 26].

Image steganographic techniques can be classified on the basis of the domains in which data is embedded. Basically there are two domains, the spatial domain and the transform domain. Steganographic techniques try to embed data in these domains.

In the spatial domain image steganography the simplest technique is to embed data in the least significant bit (LSB) of each pixel in the cover image. The LSB Replacement technique alters the insignificant information in the cover image. It places the embedding data at the least significant bit (LSB) of each pixel in the cover image. There are two types of LSB insertion methods; fixed-sized and variable-sized. The former embeds the same number of message bits in each pixel of the cover-image whereas the latter embeds a random number of bits per pixel.

Kurak was the first to present such a technique in the early nineties [Kurak and McHugh 1992]. The authors showed how one image can be hidden in another image by replacing the LSB of the cover image by the Most Significant Bit (MSB) of the hidden image.]. Another simple scheme is proposed by Chen and Lee [5,16]. The altered image is called stego-image. Altering LSB does not change the quality of image to human perception but this scheme is sensitive to a variety of image processing attacks like compression, cropping etc.

Recently, some steganographic techniques have been reported which directly modify the pixels to embed data. Some of them are reported here [27, 28, 29]. Zincheng et al. (2003) have reported a technique for reversible data hiding which has an embedding capacity of 5Kb to 60Kb and PSNR of 48dB for a 512x512x8 bit greyscale image. They embedded data by shifting the range of histograms.

Wu et al. (2003) proposed the pixel value differencing (PVD) method of steganography which can hide large amount of data by modifying the different values between pairs of adjacent pixels. Using this technique, more data can be inserted into areas where differences in the adjacent pixel values is large as pixels in these areas can tolerate more changes and this leads to good imperceptibility and a high embedding rate. Xinpeng and Shuozhong (2003) pointed out that although PVD steganography is resistant to RS steganalysis it is vulnerable to steganalysis bases on histogram of pixel value differences. Potdar et al. (2004) [19-21] showed how data can be directly embedded in the spatial domain of images by directly modifying the absolute values of pixels.

In the transform domain data can be hidden by modifying the Discrete Cosine Transform (DCT) coefficient values. These techniques are normally applicable to JPEG images because JPEG images are stored as DCT coefficient values. There are several algorithms that modify these DCT coefficient values to hide data.

The algorithm made by Derek Upham [13] was one of the first algorithm that embedded data in the frequency domain of JPEG images by modifying the DCT coefficient values. It offered an embedding capacity of 12.8% of the steganogram's size. But it was detected by chi-square test proposed by Westfeld [26]. The chi-square test proposed by Westfeld could only detect sequentially embedded messages.

Later Provos (2001) [22] proposed the Outguess algorithm to counter the statistical chi square test based on frequency counts and also offered and extended chi-square test that could detect randomly embedded messages. They also showed that their algorithm is not detected using the extended chi-square test. They observed that for JPEG images the fraction of redundant bits that can be used to hold the hidden message does not increase linearly for images with more DCT coefficients.

Another algorithm (F5) proposed by Westfeld [25] addresses the weaknesses inherent in the Outguess algorithm. This algorithm modified the absolute values of the DCT coefficients instead of modifying its LSB values. It uses matrix encoding and permutative straddling to reduce the number of steganographic changes. As a result this is resistant to the chi-square test as well as it offer more data embedding capacity compared to Outguess.

A more recent work by Sallee presents an information-theoretic method for steganography termed as Model-Based Steganography. It offers high data embedding capacity as well as resistant against statistical attacks [23].

All the techniques discussed above either try to provide either high data embedding capacity or resistance against statistical detection but we have found very few researchers who have offered breakthrough thinking in tackling the issues real time steganography and steganalysis. The main focus of this paper is to provide a framework for real time steganalysis based on the agent computing approach. In the next section we discuss the proposed conceptual framework.

3. Autonomic Agent for e-Forensics and Stego-Key Searching

The proposed conceptual framework is described in this section. The basic working of the e-Forensics methodology involves observing, identify, detecting and analyzing the image whilst preserving the message embedded in the image being examined. As such preservation is of the essence and a main focus of the e-Forensic technique. The main steps of e-Forensics pertaining to steganography are shown in Figure 3. They are:

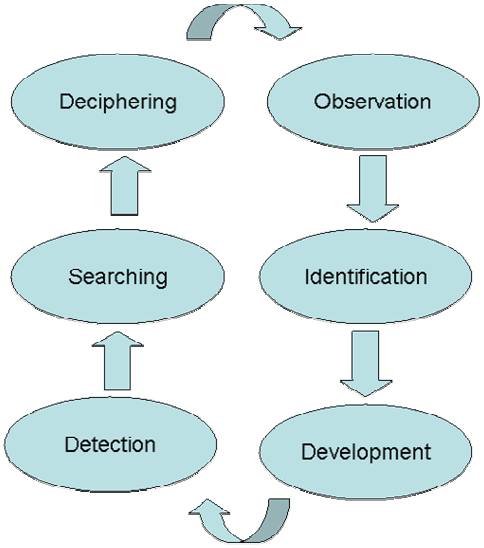


Figure 2: E-Forensics Methodology

1. *Observation* with the naked eye, with visible distortions being able to be detected, if at all possible.
2. *Identification* by means of undertaking detailed examination of the embedding mechanism
3. *Detection* of the type of steganographic software used by looking for signatures that contain repetitive patterns
4. *Development* of algorithms that can distinguish stego-images from cover images.
5. *Searching* for the stego-key and extracting the embedded data

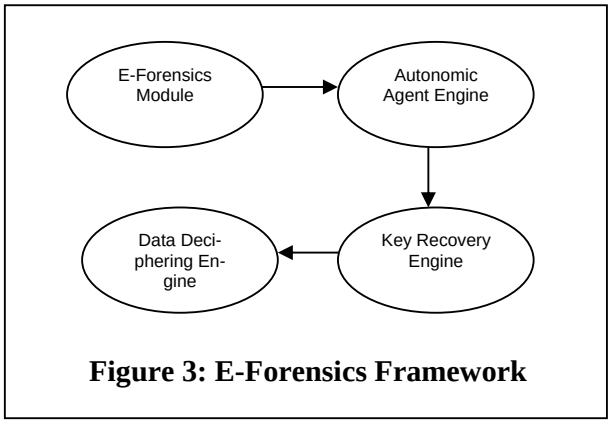


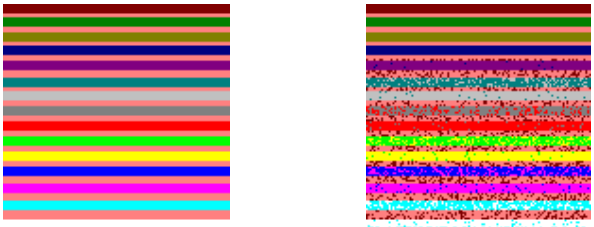
Figure 3: E-Forensics Framework

All these processes are classified into four modules; they are shown in Figure 3. The four modules are:

1. *E-Forensics Module*: This main functionality of this module is to observe, identify and develop a steganalytic strategy to recover secret messages. This module would be hosting the steganalytic algo-

rithms and a database of steganographic signatures.

2. *Autonomic Agent Engine Module*: The main functionality of this module is to carry out real time E-Forensics by visiting pre-defined web locations. The backbone of this module is Autonomic Agent Technology. We would build autonomic agent which would visit multiple hosts to randomly do e-forensic test to detect and recover secret messages.
3. *Key Recovery Engine Module*: The main function of this module is to recover key used to hide the secret message. We use the ideas presented by Fridich et al. (2004) on stego key search.
4. *Data Deciphering Engine Module*: The main function of this module is to recover the secret message based upon the successful recovery of stego key from Step 3.



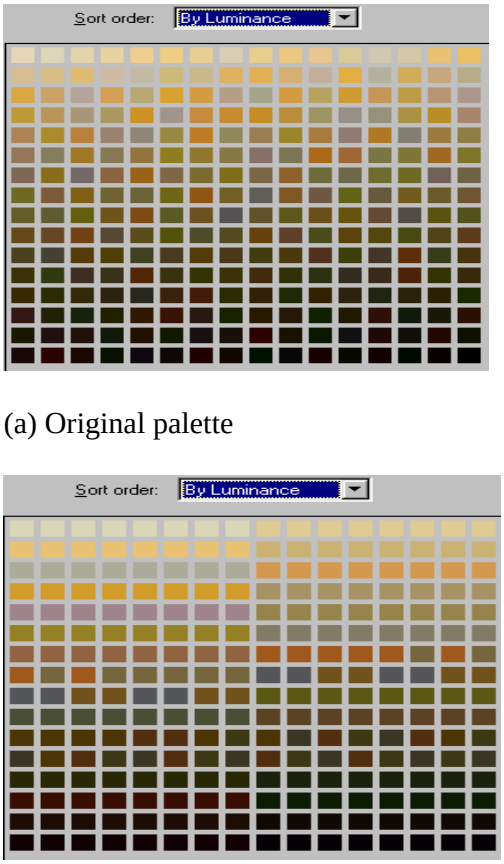
(a) 8-bit cover image (b) 8-bit stego-image

Figure 4: 'Hide and Seek' steganographic signature

Some examples of steganographic signatures are shown below. They are useful in steganalysis of image which is the core activity in E-Forensics Module. The example below, Figure 4 [14] shows palette changes using "Hide and Seek" steganographic software which produces a "noisy" image that is easily visible.

Another example [14], shown in Fig. 5 is the signature created by using S-tools steganographic software that, by sorting the palette by its luminance, reveals the signature. Although blocks of colors appear to be the same, actually they have variances of 1 bit value.

The search for the stego key which is required to extract the embedded message is of primary importance. According to Fridrich et al. [30] searches usually will follow a process of identifying recognizable structures. However if the message itself is encrypted a *dictionary attack* and/or brute-force can be used to successfully decrypt the message that is independent of the encryption algorithm.



(a) Original palette

(b) Palette distorted by 1-bit variance

Figure 5: S-tools steganographic signature

Another possible approach to search for the stego key (as suggested by Fridrich, Goljan and Soukal [35]) is to identify images that have been modified and then reverse engineer the Pseudo-Random Number Generator (PRNG) in the creation of the random path used to embed the message.

The complexity of the stego key search is determined only by the size of the stego key space and is independent of the encryption algorithm. In

more complex cases, the correct stego key can be determined through an exhaustive stego key search by quantifying statistical properties of samples along portions of the embedding path. For more details see [30].

Once the key has been identified the last step of the e-Forensics process naturally follows d*eciphering* the extracted data and obtaining the secret message (steganalysis).

4. E-Forensics System Architecture

The initial E-Forensics system is implemented on Java platform. We have used the MVC architecture when implementing the system. Each proposed module in the previous section is programmed as a separate entity. We have implemented three main components (steganalysis, key search and data deciphering) and we are currently in the process of implementing the agent based approach for real time steganalysis. The system consists of three main components: the view and user control component, logic design component and service utilities component. The detail system design is shown in Appendix A.

4.1 View and Control Component

In the e-Forensics Architecture Platform we have used view and user control object which is responsible for the Graphical User Interface of the system and managing the control functions. These two classes have the logic which is used to interact with the user and the E-Forensics system. The E_Forensics_View class is designed for displaying the image, and image characteristics. The E_Forensics_Control class is used as a control class to coordinate with the backend E-Forensics system.

4.2 E-Forensics Logic Component

E-Forensics Logic Component has seven main java classes which are used for running the E-Forensics System. This component implements all the four modules which are described in the Section 4 namely Steganalysis, Key Search, Data Deciphering, and Agent Activation. The Start_Steganalysis class implements all steganalysis algorithm as different functions and based on the Identify_Signature function a proper steganalytic algorithm is chosen. This component also implements the agent strategy to initialize and kill agents as well as getting back the responses.

4.3 E-Forensics Service and Utilities Component

E-Forensics Service and Utilities Component implements the associated classes required for image loading and adds robustness to the overall system design. These classes help the Logic Component to work efficiently. For example the class BMPFile and utility are used in manipulating for the image (.bmp) classes as java doesn't provide a ready to use class to access raw images like .bmp. FileFilters and FileOperation and ImageFileView classes are used for loading the file and setting the file filters for few files which we want to use. The rest of the classes are utils and iObserver are helping classes for loading displaying image.

5. Conclusion

In this paper we presented the architecture platform for a particular E-Forensic implementation useful in extracting information from any cover image based on the steganalytic algorithms. The novelty of our approach is the use of agent based computing to perform real time steganalysis. The proposed framework would be very effective in secret data recovery.

6. References

[1] Avcibax I., N. Memon and B. Sankur, 2003."Steganalysis using image quality metrics"

[2] Ira S. Moskowitz, LiWu Chang, Richard E. Newman 2002. Capacity is the wrong paradigm", In Proceedings of the 2002 workshop on New Security Paradigms. pp.114 – 126, Virginia Beach, Virginia

[3] Bao, F., 2002. Steganography of short messages through accessories. In Pacific Rim Workshop on Digital Steganography, July 11-12, 2002 Kitakyushu, Japan

[4] Chan, Y. & Chang, C., 2001. Concealing a Secret Image Using the Breadth First Traversal Linear Quadtree Structure. In: H. Lu, S. Spaccapietra, ed. 3rd International Symposium on Cooperative Database Systems and Applications April 23-24, 2001 Beijing, China. IEEE Computer Society, 213- 220.

[5] Chang, C., Chen, T. & Chung, L., 2002. A steganographic method based upon JPEG and quantization table modification. International Journal of Information Sciences—Informatics and Computer Science, 141(1-2), 123-138.

[6] Fisk, G., Fisk, M., Papadopoulos, C., Joshua, N., 2002. Eliminating Steganography in Internet Traffic with Active Wardens. In F.A.P. Petitcolas, ed. 5th International Workshop on In Information Hiding October 7-9, 2002 Noordwijkerhout, The Netherlands. Springer, 18-35.

[7] Fridrich, J., "Feature Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes" 2004. In Proc. 6th Information Hiding Workshop, Toronto, Canada, May 23-25, 2004.

[8] Fridrich, J., Goljan, M., Hogea, D., Soukal, D., "Quantitative Steganalysis of Digital Images: Estimating the Secret Message Length" 2003. In ACM Multimedia Systems Journal, Special issue on Multimedia Security, Vol. 9(3), 2003, pp. 288-302

[9] Fridrich, J., Goljan, M., Hogea, D., 2002. "Attacking the Outguess".

[10] Fridrich, J., Goljan, M., Hogea, D., 2002. "Steganalysis of JPEG Images: Breaking the F5 Algorithm".

[11] Fridrich, J., Goljan, M., Du R., 2000. "Steganalysis based on JPEG Compatiblity Steganalysis"

[12] Hsu, C. T., Wu, J. L., 1999. Hidden Digital Watermarks in Images. In IEEE Transactions on Image Processing, 8(1), 58-68.

[13] Derek Upham, 1999. "Jsteg Steganographic Algorithm" Available on the internet ftp://ftp.funet.fi/pub/crypt/steganography/

[14] Johnson, N. F., Jajodia, S., 1998. Steganalysis of images created using current steganographic software. In: D. Aucsmith, ed. 2nd International Workshop on Information Hiding April 14-17, 1998, Portland, Oregon, USA. Springer, 273-289.

[15] Khan, M., Potdar V, Chang E., 'A prototype implementation of Grey Level Modification Steganography', Accepted in 33rd International Conference Korea 2004.

[16] Lee, Y. K. & Chen, L. H., 2000. High Capacity Image Steganographic Model. In IEE Proceedings Vision, Image and Signal Processing. 147(3), 288-294.

[17] Newman, R. E., Moskowitz, I. S., Chang, L., Brahmadesam M. M., 2002 " A Steganographic Embedding Undetectable by JPEG Compatibility Steganalysis". In: Petitcolas (Ed.): Information Hiding, 5th International Workshop, IH 2002, Noordwijkerhout, The Netherlands, October 7-9, 2002 LNCS Springer Verlag 258-277.

[18] Petitcolas, F. A. P., Anderson, R. J. & Kuhn, M. G., 1999. Information Hiding — A Survey. In Proceedings of the IEEE, 87(7), 1062-1078.

[19] Potdar V, Chang E. 'Covering Encrypted Information using Images', European and Mediterranean Conference on Information Systems (EMICS2004), Tunis, Tunisia, July 25-27, 2004

[20] Potdar V, Chang E. 'Hiding Text Cryptography using Image Cryptography', 4th International Networking Conference, Plymouth, U.K. July 6-9, 2004

[21] Potdar V, Chang E. 'Grey Level Modification Steganography for Secret Communication', 2nd IEEE International Conference on Industrial Informatics (INDIN2004), Berlin, Germany, June 24-26, 2004

[22] Provos, N., "Defending Against Statistical Steganalysis" 2001. In Proceedings of the 10th USENIX Security Symposium, pages 323-335, August 2001

[23] Sallee, P., "Model-Based Steganography" 2003. In International Workshop on Digital Watermarking, Seoul, 2003,

[24] Simmons, G. J., 1984. " The prisoner's problem and the subliminal channel" In Advances in Cryptology -- CRYPTO '83", D. Chaum, ed., Plenum Press, 1984, 51-67.

[25] Westfeld, A., 2001. "High Capacity Despite Better Steganalysis (F5-A Steganographic Algorithm)", In: Moskowitz, I.S. (eds.): 4th International Workshop on Information Hiding, LNCS, Vol. 2137. Springer-Verlag, New York, pp. 289--302, 2001.

[26] Westfeld, A., Pfitzmann A., 2000. "Attacks on Steganographic Systems Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools—and Some Lessons Learned". Lecture Notes in Computer Science, vol.1768, Springer-Verlag, Berlin, 2000, pp.

[27] Wu, D.C. and Tsai, W.H., 2003. A steganographic method for images by pixel-value differencing. In Pattern Recognition Letters. 24(9-10), 1613-1626.

[28] Xinpeng, Z., Shuozhong, W., 2003. Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. In Pattern Recognition Letters. 25(3), 331-339

[29] Zincheng, N., Yun, Q. S., Ansari, N., Wei, S., 2003. Reversible Data Hiding.

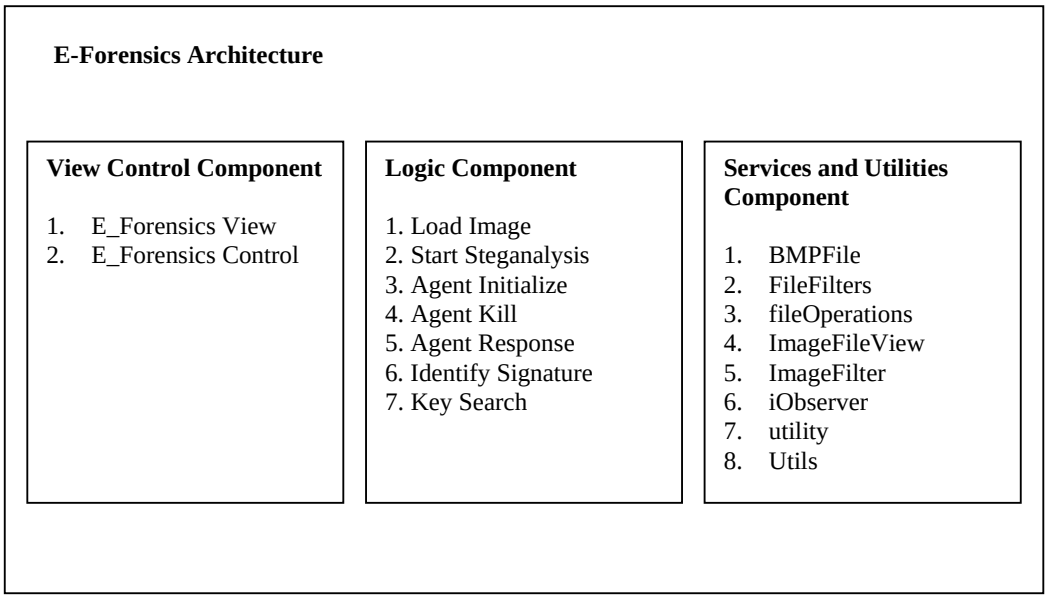
[30] J. Fridrich, M. Goljan and D. Soukal, "Searching for the Stego Key", Proc. EI SPIE San Jose, CA, Jan 2004. PDF

[31] Walter, Peter., Chambers dictionary of science and technology, Edinburgh: Chambers.

[32] Healy, R (2004), "Using electronic evidence from proprietary devices: Opportunities and implications for court evidence" 17th International Symposium on Forensic Sciences, Wellington, New Zealand

[33] McKemmish, R (1999) http://www.usg.edu/oiit/re/re03/proceedings/forensics.pdf

Appendix A


E-Forensics Architecture
View Control Component
1. E_Forensics View
2. E_Forensics Control
Logic Component
1. Load Image
2. Start Steganalysis
3. Agent Initialize
4. Agent Kill
5. Agent Response
6. Identify Signature
7. Key Search
Services and Utilities Component
1. BMPFile
2. FileFilters
3. fileOperations
4. ImageFileView
5. ImageFilter
6. iObserver
7. utility
8. Utils

Appendix B

Example of Agent based e-Forensics for Extracting Secret Information

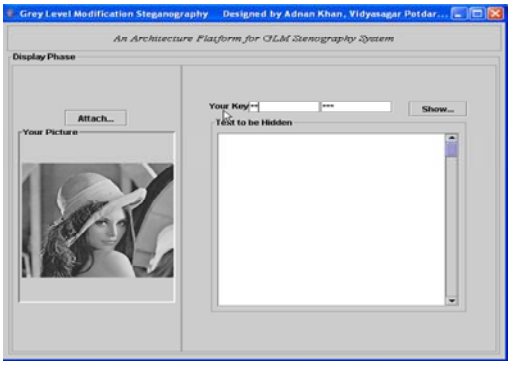


Figure 6 (a). Interface for image selection and searching key

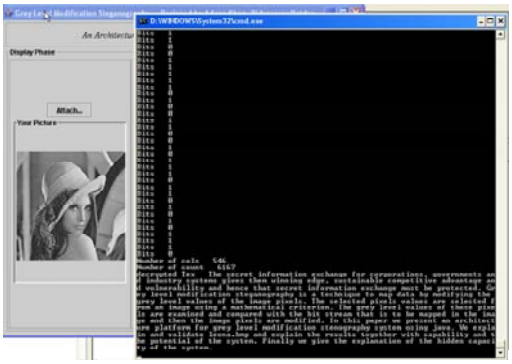


Figure 6 (b). Backend log file showing the extraction process.

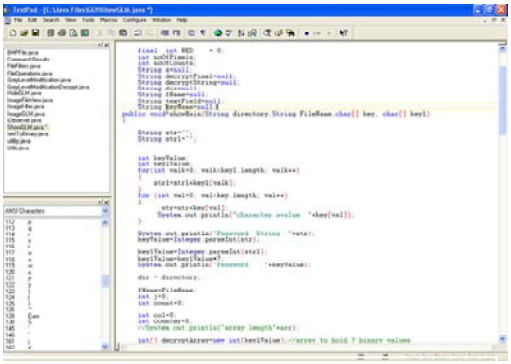


Figure 6 (c). Screenshot of the implantation code

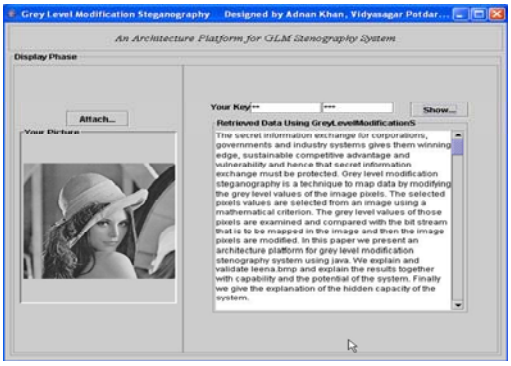


Figure 6 (d). Based on the key generated data can be recovered as shown above.