

Case Study: An Implementation of a Secure Steganographic System

Xuan-Hien Dang

Department of Computer Science
University of Akron
Akron OH, USA

Krishna C. S. Kota

Department of Computer Science
University of Akron
Akron OH, USA

Abstract - *In this paper, we present the development of a Windows-based application implemented on the Microsoft .NET platform as a framework to allow a user to first compress and encrypt a secret message then hide its content into a cover image or audio file as well as extracting and retrieving the original data. A secure and robust steganographic system is achieved by applying and combining compression, encryption and steganography techniques.*

Keywords: Steganography, Information hiding, Security.

1 Introduction

Today's computing systems, which consist of a broad range of processors, communication networks and information repositories, are vital to the operation of many sectors of our society, from financial and manufacturing to education and healthcare. The explosive growth and the open nature of the Internet and e-commerce have caused organizations to become more vulnerable to malicious electronic attacks than ever before. With the increasing quantity and sophistication of attacks on IT assets, companies have been suffering from breach of data, loss of customer confidence and job productivity degradation, all of which eventually lead to loss of revenue. Many network security measures have been proposed to counter those attacks in order to guarantee the confidentiality, integrity, authenticity and availability of resources [1-6].

In this paper, we consider the problem of protecting transmitted data to ensure its privacy using the combination of cryptography and steganography techniques. On one hand, with cryptography, secret messages are converted into a format that is incomprehensible and unreadable without the knowledge of secret information. On the other hand, with steganography, the secret

message is concealed into a host medium such as text, image, audio or video, so that the hidden data are imperceptible from unintended observers. The main advantage of steganography is its ability to conceal the mere existence of hidden data, thus preventing an observer from selectively blocking the transmission of such data. Two important properties of a steganographic system are the amount of secret data, called the data payload, and the imperceptibility of the data. However, a large amount of hidden data will necessarily increase its perceptibility. In this work, we have developed a tool that allows a user to hide secret data in an image by successively applying compression to reduce the size of the hidden message, encryption and steganography techniques. Section 2 provides a quick background on the different methods for compression, encryption and steganography. Section 3 discusses the implementation of the tool that integrates those security techniques to ensure a higher level of security for the secret data. Finally section 4 offers future work and conclusions.

2 Background and Related Work

In an image-hiding system, the image used to embed secret data is referred to as the host or cover image. The resultant image, which is embedded with secret data, is called the stego-image and is expected to exhibit the same content as the host image. Many image steganography methods for hiding data in images have been proposed [1-4], but typically two types of image hiding schemes are often used.

The first approach is based on the spatial domain of the cover image and relies on least-significant bit (LSB) substitution, which embeds the secret data directly into the pixels of the cover image using some mapping rules.

The second approach is based on the frequency domain of the cover image which makes use of transformation functions such as the discrete Fourier transform to map pixel values in the spatial domain into the frequency domain. Though the hiding capacity in this type of approach is limited, it offers better robustness.

Steganography and cryptography complement each other and are generally used together to achieve maximal security. Many cryptographic techniques have been proposed [5-6] and are essentially based on keys and associated algorithms used for encrypting and decrypting data. There are two main categories of cryptosystems: secret key or symmetric and public key or asymmetric. Symmetric cryptosystems use the same key (the secret key) to encrypt and decrypt a message, whereas asymmetric cryptosystems use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it. Several secret key cryptography algorithms commercially in use today include the Data Encryption Algorithm (DES), the Advanced Encryption Standard (AES) and the International Data Encryption Algorithm (IDEA). One popular public-key cryptography algorithm is the Rivest, Shamir and Adleman (RSA) algorithm [6]. It has been used for key exchange, digital signatures and encryption of small blocks of data.

Among the basic requirements of a mature steganographic system are the capacity and security requirements. For capacity, it is required to keep the amount of secret data as much as can be embedded in a given cover data. A large data payload becomes easily detectable. To obtain higher payloads, one can make use of compression techniques. Data compression is the process of encoding information using fewer bits than a more obvious representation would use, through the application of specific encoding schemes. Data can either be lossless compressed or lossy compressed. The original data can be perfectly recovered from the lossless-compressed data, since no data is lost during the compression process. Contrarily, some amount of data is lost in the lossy compression and the recovered information is an approximate version of the original data. For the security requirement, it is fulfilled in general implementation by either the encryption manipulation using private-key/public-key system or through pseudo-random generation using a seed key. Assuming that an encryption method

produces messages that appear random without the proper key and that the intended receiver has the same compressor and decompressor, a robust and secure steganographic system can be achieved.

Accordingly, in this work, we have developed a Windows-based framework implemented on the Microsoft .NET platform for a secure information hiding system which allows the user to conceal and extract a secret message that is first compressed and then hidden into a cover image.

3 Implementation

The objectives of this work are to design and implement a steganographic system which integrates a compression module and an encryption module to improve its capacity and security requirements. We discuss here the different phases involved in such implementation.

To embed secret data into a cover image requires two files. The first is the *cover file or cover image* that holds the hidden information. The second is the *message* – the information to be hidden. A message may be plain text, cipher text, other images or anything that can be embedded in a bitstream. When combined, the cover file and embedded message become a *stego-file*. A *stego-key* (a type of password) may be used to hide data, and then later used to decode the message.

Similarly to cryptography, secret key steganography requires the exchange of a secret key (stego-key) prior to communication. Secret Key Steganography takes a cover medium and embeds the secret message using the stego-key. Only the intended parties are able to reverse the process and read the secret message. In public key steganography, the sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message. The public key approach provides a more robust way of implementing a steganographic system because it can utilize a much more robust and researched technology. It has multiple levels of security in that unwanted parties must first suspect the use of steganography and then would have to find a way to crack the algorithm used by the public key system before they could intercept the secret message. Figure 1 shows the overall structure of the steganographic system where f_E denotes the steganographic embedding function and f_E^{-1} the steganographic extracting function.

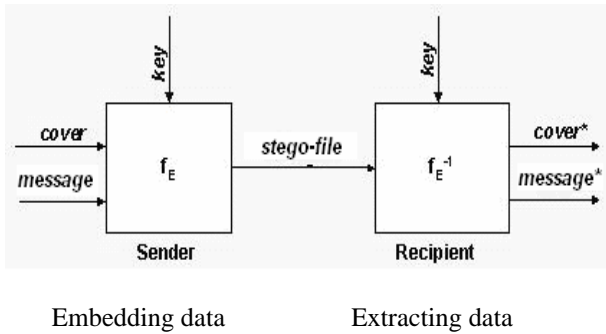


Figure 1. Overall structure of the steganographic system

The steganographic system allows the encoding and hiding of secret messages in cover images or audio files. The encryption and decryption algorithms implemented are the DES and the RSA algorithms using C# .NET libraries. Lossless and lossy compressions are also offered. The complete process which involves compression, encryption and hiding information is shown in Figure 2.

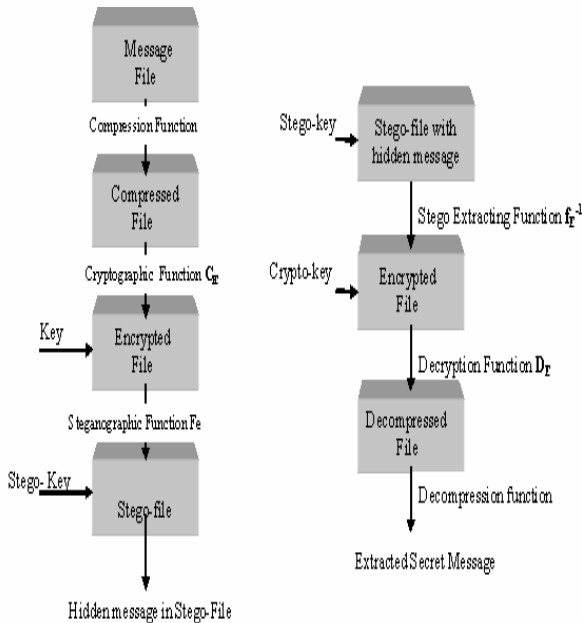


Figure 2. Embedding and Extracting Procedures

The application developed consists of a user-friendly interface that allows the user to select the different techniques separately or as a single combined security module as shown in Figure 3. It allows two types of data to be used as cover: image and audio data as shown in figures 4 and 5.

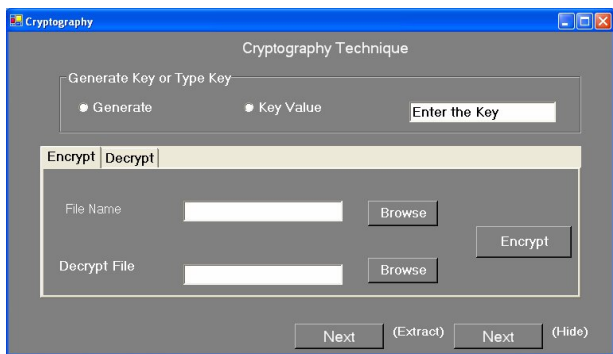
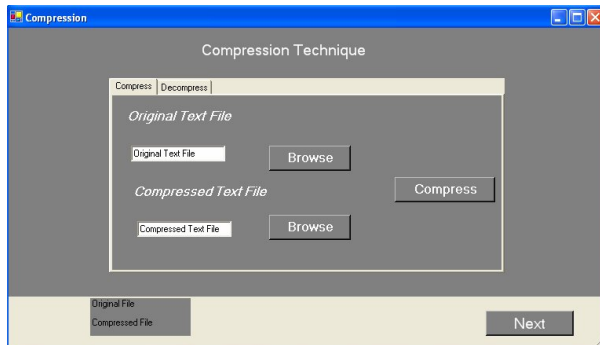
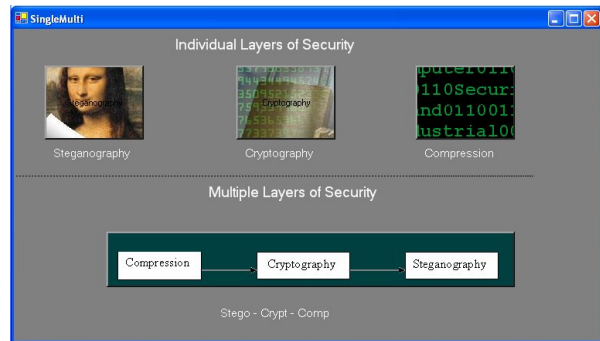
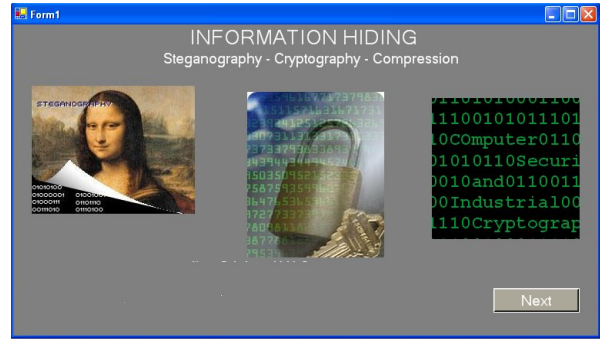


Figure 3. Graphical User Interface for the steganographic system

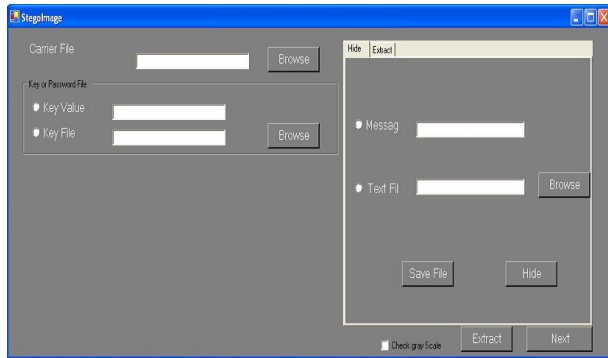


Figure 4. Steganography using an image file

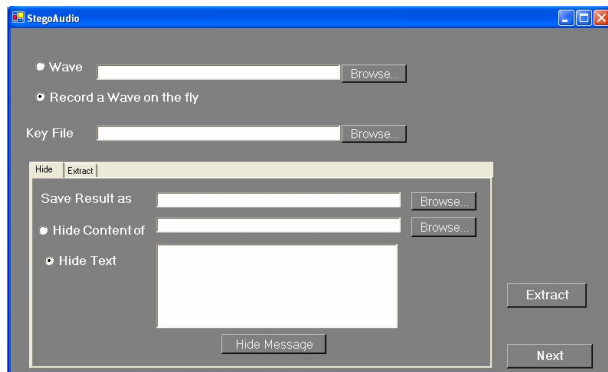


Figure 5. Steganography using an audio file

4 Conclusions

In this paper, we have developed a Windows-based application implemented on the Microsoft .NET platform that allows a user to encrypt a secret message and hide its content into a cover image or audio file as well as extracting and retrieving the original data. Several compression, encryption and steganography techniques were combined and successfully implemented in order to provide a secure steganographic system with different layers of security. Future work will include a broader variety of steganographic techniques.

5 References

- [1] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security & Privacy* Vol 1, No. 3, pp. 32-44, May/June 2003.
- [2] D. Artz, "Digital Steganography: Hiding data within data," *IEEE Internet Computing* Vol 5, No. 3, pp. 75-80, May/June 2001.

[3] N. F. Johnson, Z. Duric, S. Jajodia, *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.

[4] Y.-H Yu, C.-C. Chang and Y.-C. Hu, "Hiding secret data in images via predictive coding", *Pattern Recognition* Vol 38, No. 5, pp. 691-705, 2005.

[5] B. Schneier, *Applied Cryptography*, second edition, Wiley, New York, 1996.

[6] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Commun. ACM* Vol 21, No. 2, pp. 120-126, February 1978.