

IMAGE STEGANOGRAPHY APPLICATIONS FOR SECURE COMMUNICATION

by

Tayana Morkel

Submitted in partial fulfillment of the requirements for the degree
Master of Science (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

May 2012

Image Steganography Applications for Secure Communication

by

Tayana Morkel

E-mail: tmorkel@cs.up.ac.za

Abstract

To securely communicate information between parties or locations is not an easy task considering the possible attacks or unintentional changes that can occur during communication. Encryption is often used to protect secret information from unauthorised access. Encryption, however, is not inconspicuous and the observable exchange of encrypted information between two parties can provide a potential attacker with information on the sender and receiver(s). The presence of encrypted information can also entice a potential attacker to launch an attack on the secure communication.

This dissertation investigates and discusses the use of image steganography, a technology for hiding information in other information, to facilitate secure communication. Secure communication is divided into three categories: self-communication, one-to-one communication and one-to-many communication, depending on the number of receivers. In this dissertation, applications that make use of image steganography are implemented for each of the secure communication categories. For self-communication, image steganography is used to hide one-time passwords (OTPs) in images that are stored on a mobile device. For one-to-one communication, a decryptor program that forms part of an encryption protocol is embedded in an image using image steganography and for one-to-many communication, a secret message is divided into pieces and different pieces are embedded in different images. The image steganography applications for each of the secure communication categories are discussed along with the advantages and disadvantages that the applications have over more conventional secure communication technologies. An additional image steganography

application is proposed that determines whether information is modified during communication.

Keywords: computer security, information hiding, steganography, image processing, secure communication, image authentication

Supervisor : Prof. A.P. Engelbrecht
Department : Department of Computer Science
Degree : Master of Science

Acknowledgements

I would like to take this opportunity to thank the following people who supported me in the completion of this dissertation:

- My family for their neverending encouragement. I would not have been able to complete this dissertation without their support and motivation. A special thank you to my husband, Lourens, for his constant love and support (and proof reading).
- My supervisor, Prof. Andries Engelbrecht, for his insight, understanding and feedback.

TABLE OF CONTENTS

LIST OF FIGURES	x
LIST OF TABLES	xii
CHAPTER 1: INTRODUCTION	1
1. INTRODUCTION	1
2. SECURE COMMUNICATION	3
3. RESEARCH OBJECTIVES	4
4. CONTRIBUTIONS TO THE FIELD	5
5. CHAPTER LAYOUT	5
CHAPTER 2: STEGANOGRAPHY AS AN ALTERNATIVE TO CRYPTOGRAPHY	7
1. INTRODUCTION	7
2. COMPARISON MEASURES	7
3. CRYPTOGRAPHY	9
3.1 Definition of cryptography	9
3.2 Traditional uses of cryptography	10
3.3 Encryption algorithms and the cryptographic key	11
3.4 Security services offered by cryptography	11
3.5 Encryption problems	12
4. STEGANOGRAPHY	14
4.1 Definition of steganography	15
4.2 Traditional uses of steganography	17
4.3 Steganography algorithms and the steganographic key	18
4.4 Security services offered by steganography	18
4.5 Steganography problems	19
5. CRYPTOGRAPHY VS. STEGANOGRAPHY	20
6. CONCLUSION	22
CHAPTER 3: CATEGORISATION OF STEGANOGRAPHY	23
1. INTRODUCTION	23
2. CATEGORISATION ACCORDING TO STEGANOGRAPHIC TECHNIQUES	23

3.	CATEGORISATION ACCORDING TO CARRIER TYPES	25
3.1	Text Steganography	26
3.2	Image Steganography	26
3.3	Audio/Video Steganography	27
3.4	Protocol Steganography	28
4.	CONCLUSION	28
CHAPTER 4: DIGITAL IMAGES AND COMPRESSION		29
1.	INTRODUCTION	29
2.	DIGITAL IMAGING CONCEPTS	29
2.1	Colour representation	29
2.2	Image definition	30
3.	IMAGE COMPRESSION	31
3.1	Lossless compression	32
3.2	Lossy compression	32
3.3	Compression and steganography	32
4.	IMAGE FILE FORMATS	33
4.1	Spatial domain formats	33
4.1.1	Raster images	33
4.1.2	Palette based images	35
4.2	Transform domain formats	36
5.	CONCLUSION	38
CHAPTER 5: IMAGE STEGANOGRAPHY		39
1.	INTRODUCTION	39
2.	EVALUATION CRITERIA	40
3.	SPATIAL DOMAIN STEGANOGRAPHY	41
3.1	Raster images	41
3.1.1	Overview of LSB embedding	41
3.1.2	Weaknesses of LSB embedding	42
3.1.3	Improvements to LSB embedding	43
3.2	Palette based images	43

3.2.1	LSB embedding in palette based images	44
3.2.2	Weaknesses of LSB embedding in palette based images	45
3.2.3	Optimal parity embedding	45
3.	TRANSFORM DOMAIN STEGANOGRAPHY	46
4.1	JPEG steganography	46
4.2	Weaknesses of JPEG steganography	47
4.3	Outguess	47
4.4	F5	48
5.	EVALUATION OF THE IMAGE STEGANOGRAPHY ALGORITHMS	48
5.1	Invisibility	48
5.2	Payload capacity	49
5.3	Robustness against image manipulation attacks	49
5.4	Statistical undetectability	50
5.5	Summary of image steganography algorithm comparison	50
6.	CONCLUSION	50
 CHAPTER 6: SELF-COMMUNICATION		52
1.	INTRODUCTION	52
2.	ONE-TIME PASSWORDS	53
3.	OVERVIEW OF THE STEGO-OTP SYSTEM	56
4.	DESIGN OF THE STEGO-OTP SYSTEM	57
4.1	Generating the OTP list	58
4.2	Selecting an image steganography algorithm	58
4.3	Selecting an image	59
4.4	Extracting the image	61
5.	EVALUATING THE STEGO-OTP SYSTEM	63
6.	CONCLUSION	64
 CHAPTER 7: ONE-TO-ONE COMMUNICATION		65
1.	INTRODUCTION	65
2.	DECRYPTOR DISTRIBUTION	66
3.	THE DECRYPTOR DISTRIBUTION SYSTEM	67

3.1	The embedding phase	68
3.1.1	Algorithms used in the embedding phase	69
3.1.2	The decryptor format	70
3.1.3	The decryptor header	71
3.1.4	LSB embedding algorithm	73
3.2	The extraction phase	74
3.2.1	Stego-extraction algorithm used in the extraction phase	74
3.2.2	The LSB-extraction algorithm	75
4.	DECRYPTOR DISTRIBUTION PROTOTYPE	76
4.1	Embedding the decryptor	76
4.2	Embedding the encrypted message	77
4.3	Extracting the decryptor	78
5.	EXPERIMENTAL RESULTS TO DETERMINE INVISIBILITY OF EMBEDDED INFORMATION	78
6.	CONCLUSION	81
CHAPTER 8: ONE-TO-MANY COMMUNICATION		82
1.	INTRODUCTION	82
2.	EXISTING GROUP COMMUNICATION TECHNOLOGIES	83
3.	OVERVIEW OF THE MESSAGE DISTRIBUTION SYSTEM	85
4.	SECRET SHARING	86
4.1	Shamir's secret sharing scheme	86
4.2	Related work on Shamir's secret sharing scheme	87
4.3	Shamir's secret sharing scheme in the message distribution system	88
5.	TECHNICAL DETAILS OF THE MESSAGE DISTRIBUTION SYSTEM	90
5.1	Selecting an image steganography algorithm	91
5.2	Selecting the images and the websites	91
5.3	Distributing list of websites to receivers	92
6.	EVALUATING THE MESSAGE DISTRIBUTION SYSTEM	92
7.	CONCLUSION	93

CHAPTER 9: RECURSIVE IMAGE STEGANOGRAPHY FOR DATA INTEGRITY	94
1. INTRODUCTION	94
2. RELATED TECHNOLOGIES	96
3. THE RECURSIVE IMAGE STEGANOGRAPHY SYSTEM	99
4. DESIGN OF THE RECURSIVE IMAGE STEGANOGRAPHY SYSTEM	100
4.1 Selecting an image steganography algorithm	101
4.2 Recursive image steganography	102
4.3 Image selection	102
4.4 Displaying the inner image	104
4.5 The recursive image steganography prototype	105
5. EXPERIMENTAL RESULTS	105
5.1 Significant changes	108
5.2 Insignificant changes	110
6. CONCLUSION	113
CHAPTER 10: CONCLUSION	115
1. SUMMARY	115
2. FUTURE RESEARCH	116
BIBLIOGRAPHY	118
APPENDIX A: PUBLICATIONS DERIVED FROM THIS WORK	129

LIST OF FIGURES

2.1	Communication over an insecure channel	9
2.2	A model of the steganographic process	15
3.1	Categories of steganography based on carrier types	25
4.1	Pixels and bit representation of a greyscale image with bit depth 8	30
4.2	Pixels and bit representation of a 24-bit colour image using the RGB colour model	31
4.3	Pixels and indexes of an 8-bit GIF image with image palette	35
4.4	RGB conversion and UV downsampling	37
4.5	The discrete cosine transform (DCT) process	38
6.1	The Stego-OTP system	57
6.2	A scenario where the image is transmitted through an untrusted network to a secure server. The application on the server extracts the password and sends it back to the workstation	62
6.3	A scenario where the client workstation issues a request to the secure server. The server transmits the extraction application to the client. The image and password is never communicated over the network	62
7.1	Process diagram of the decryptor distribution system	68
7.2	Representation of information embedded in cover image	73
7.3	UML Class diagram of decryptor distribution prototype	77
7.4	Process diagram for decryptor distribution prototype with detailed information .	79
7.5	Comparison of 24-bit colour cover image, with stego image containing 2,702 bytes of embedded information	79
7.6	Comparison of 8-bit greyscale cover image, with stego image containing 2,702 bytes of embedded information	80
7.7	Comparison of 4-bit greyscale cover image, with stego image containing 2,702 bytes of embedded information	81
8.1	Process diagram of the message distribution system	85
9.1	Recursive image steganography process	99

9.2	Representation of information distribution in inner image and stego image during recursive image steganography	103
9.3	GUI of embedding phase of recursive image steganography prototype	106
9.4	Stego image for experimental purposes	107
9.5	Inner image for experimental purposes	107

LIST OF TABLES

2.1	Summary of differences between watermarking, fingerprinting and steganography	17
2.2	Comparison of cryptography and steganography	20
3.1	Steganography technique categories	24
5.1	Comparison of image steganography algorithms	51
9.1	Experimental results of significant changes and image manipulation techniques performed on stego image	108
9.2	Experimental results from altering eight individual, randomly selected bytes of the stego image	111
9.3	Experimental results from altering eight consecutive bytes of the stego image ..	112
9.4	Experimental results from altering eight consecutive bytes of the stego image at the precise location of the embedded message	113

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Communication of secret information is a critical factor in information technology that continues to create challenges with increasing levels of sophistication. When communication takes place between parties that are located on the same secure network, these challenges can be considered as manageable. However, in the modern era expectations are that one can travel the world and receive secret information at the same time without jeopardising the confidentiality of secret information. In these situations where the involved parties are spatially separate, the security of secret information cannot rely only on the advanced technologies of secure networks, and additional security mechanisms should be incorporated.

Since it is unlikely to have a dedicated network line spanning the width of the globe, communication that takes place between remote users often relies on existing public infrastructure, particularly the Internet. Public channels such as the World Wide Web (www) and e-mail are convenient to use for remote communication, with availability being the main advantage. However, the uncertainty of acceptable security is the most significant disadvantage of these public channels. In some scenarios virtual private networks (VPNs) can be implemented to facilitate secure communication by acting as a private tunnel between two parties (Conklin et al 2004:266-7). However, even in these circumstances a VPN still makes use of the Internet to establish the network, and once an intruder gains access to a VPN tunnel, he has access to the entire network (Mavrakis 2003:12).

Internet and web-based systems are vulnerable to a variety of well-known cyber-attacks, including denial-of-service attacks, spoofing, and many more (Jefferson et al 2004:60). A man-in-the-middle attack particularly focuses on intercepting communication, mainly between a client and a server (Schneier 1963:114). By placing himself between the client and server, an attacker can ensure that all communication between two parties passes through him first, thereby allowing him to read, modify, inject, or drop any communication packet (Xia & Brustoloni 2005:489).

To communicate over an insecure channel, cryptography has been developed as a technique for constructing a secure logical channel over an insecure physical channel (Gollmann 1999:201). Many different cryptographic techniques have been developed to encrypt and decrypt data by scrambling the information in order to secure it.

Cryptography, however, suffers from a number of drawbacks – notably the fact that the mere presence of an encrypted message might be cause for suspicion in itself (Shirali-Shahreza & Shirali-Shahreza 2006:316-21; Kawaguchi & Eason 1999:464-73). If an eavesdropper should intercept an encrypted message, he might argue that the information must be valuable, since someone went through the trouble of encrypting it in the first place.

Another drawback of cryptography is the limitations that have been enforced by certain governments, which is particularly significant when cryptography is to be used by remote users. Many governments have created laws to either limit the strength of cryptographic systems or to prohibit it altogether (Krinin 2000; Grodzinsky, Miller & Wolf 2007:205). Primarily due to law enforcement's fear of not being able to gain intelligence by information interception (Grodzinsky, Miller & Wolf 2007:205), the use of cryptography – and sometimes even the possession of a cryptographic algorithm – is illegal in certain countries (Dunbar 2002:3).

Although researchers are constantly developing improvements to current cryptographic systems, the potential cause for suspicion and legal limitations are inherently part of the way that cryptographic system's function, and cannot generally be improved upon. Alternative mechanisms that could improve upon these limitations should thus be investigated. Steganography is one such mechanism that attempts to protect sensitive information from unauthorised parties.

Steganography is a technology that is used to hide secret information in digital media, thus hiding the fact that secret communication is taking place (Jamil 1999:10). By hiding secret information in less suspicious digital media, well-known channels, for example e-mail and social networking sites, are avoided, thereby reducing the risk of information being leaked in transit (Artz 2001:75). Should an attacker attempt to intercept the communication through a man-in-the-middle attack, he would have no reason to suspect that he has intercepted anything more than an innocent image, for example.

Steganography can be used to enhance the security of various applications, including secure communication. Different approaches to secure communication, as discussed in the next section, entail different implementations of steganography.

2. SECURE COMMUNICATION

When referring to computer-mediated communication, communication is defined as the means of sending and receiving information (Oxford 2005), specifically from one computer or device to another (Webopedia). In the context of this dissertation, secure communication is defined as sending and receiving information with the certainty that the information remains safe and protected against attacks.

For the purposes of this definition and in order to improve upon the limitations of a cryptographic system, requirements for a secure communication system are as follows:

1. The fact that secret information is being communicated should be concealed and communication should take place in an inconspicuous manner.
2. The confidentiality of secret information should be ensured, even under the suspicion that secret information is being communicated.
3. The communication should allow the user to comply with international laws regarding the use of cryptography.
4. The communication should be done (almost) as easily as it would have been using traditional secure communication systems and should be convenient to use by non-technical users.

At first glance, a steganographic system complies with all of the above requirements. However, in order to avoid premature conclusions about the level of compliance that steganography provides, different secure communication scenarios should be investigated. To address different secure communication scenarios, communication is divided into categories according to the number of involved parties. Steganography will then be implemented for each category to determine if it does comply with the requirements and can be used as an alternative to cryptography.

Probably the most common type of communication is where one sender communicates information to one receiver – hence known as one-to-one communication. Closely related to this is one-to-many communication, where one person communicates information to a group of individuals. It is important to note that this is not the same as where one entity, for example a service provider, communicates information such as passwords to its clients, since then the secret information sent to each client is unique. For the purpose of the research proposed in this dissertation, each receiver in a one-to-many communication receives the same secret information.

Perhaps not always recognised as communication in the strict sense of the word, is the ability to communicate with yourself. The scenario where a user stores information, for example a password, somewhere where it can be retrieved and used at a later stage by the user, can be considered as self-communication. A wallet would be an appropriate example of such a scenario where the owner can store information in the wallet to be found and used later.

Secure communication can thus be refined to consist of three categories: secure self-communication, secure one-to-one communication and secure one-to-many communication. The problem to be examined in this dissertation is thus to determine if steganography can be applied to different communication systems in order to comply with the requirements of inconspicuousness, confidentiality, legality, and usability.

3. RESEARCH OBJECTIVES

Simplified, the problem to be addressed in this dissertation is to communicate secret information to remote users over an insecure channel. However, when considering the limitations of cryptography, cryptographic techniques cannot be used to solve this particular problem and the application of steganography, as an alternative solution, is explored. Secure communication, can also be divided into categories and the different categories of secure communication have different requirements and problems.

The main objective of this dissertation is thus to study the application of image steganography to facilitate secure communication in self-communication, one-to-one communication and one-to-many communication. In reaching this objective, the following sub-objectives were identified:

- To compare steganography with cryptography to determine whether steganography is a suitable alternative to cryptography for secure communication.
- To provide an overview of existing image steganography algorithms and discuss the strong and weak points offered by each of the algorithms.
- To briefly discuss digital image formats and compression in the spatial and transform domain to enable a better understanding of how information can be embedded in images.
- To investigate existing technologies used to facilitate secure self-communication, one-to-one communication and one-to-many communication, and determine why they do not comply with all of the requirements of a secure communication system.
- To study the combination of image steganography with other security technologies, such as one-time passwords and secret sharing schemes
- To investigate an application of image steganography that determines whether information that was communicated was modified during communication.

4. CONTRIBUTIONS TO THE FIELD

This dissertation proposes the use of image steganography instead of cryptography when confidentiality is required during secure communication. Secure communication is divided into self-communication, one-to-one communication and one-to-many communication and for each of these communication scenarios this dissertation investigates the adaptation and application of image steganography.

This dissertation contributes to the field through a comparison between steganography and cryptography based on the security services offered by each technology. The development of three image steganography applications for the three secure communication categories are further contributions. Another contribution is the implementation of a mechanism for using image steganography for image authentication.

5. CHAPTER LAYOUT

The remainder of the dissertation is organised as follows:

- **Chapter 2:** A brief overview of cryptography and steganography is given and a comparison is made between the two technologies based on the ISO 7498-2 (1989) security services that they offer. The vulnerabilities and possible attacks against cryptography and steganography are briefly discussed.
- **Chapter 3:** Categorisation of image steganography is done, first according to steganographic technique and then according to carrier types.
- **Chapter 4:** A brief overview of digital images is given, including colour representation and image definition. Digital images are divided into spatial domain formats and transform domain formats and the image representation in each domain is discussed along with the compression techniques used in each domain.
- **Chapter 5:** Different image steganography algorithms in the spatial domain and transform domain are discussed. Evaluation criteria are proposed to define the requirements of a secure image steganography algorithm. Each of the discussed algorithms is evaluated according to the criteria.
- **Chapter 6:** Self-communication is discussed along with existing technologies used for secure self-communication. A system is proposed that uses image steganography to hide one-time passwords in images on a mobile device.
- **Chapter 7:** One-to-one communication is discussed along with the vulnerabilities of secure one-to-one communication and the technologies that are currently used to secure the communication. A system is proposed that uses image steganography to hide a computer program in an image.
- **Chapter 8:** One-to-many communication is discussed. A system is proposed that divides secret information into pieces using Shamir's secret sharing scheme and hides the secret pieces in images on the Internet.
- **Chapter 9:** Discusses the integrity of communicated information and the existing technologies used for ensuring data integrity. An application of image steganography is proposed that uses steganography recursively to visually determine if communicated information was modified en route.
- **Chapter 10:** The dissertation conclusion is given and ideas for future research are discussed.

CHAPTER 2

STEGANOGRAPHY AS AN ALTERNATIVE TO CRYPTOGRAPHY

1. INTRODUCTION

One of the objectives given in chapter 1 was to determine whether steganography could be seen as a suitable alternative to cryptography. This chapter discusses cryptography and steganography and suggests comparison criteria for comparing steganography and cryptography.

In order to perform the comparison, background information on each technology is first given. Steganography is discussed in more detail, since the focus of this dissertation is on steganography and not on cryptography, although basic background information on cryptography is also provided. Strong and weak points of both steganography and cryptography are examined and the differences and similarities between the two technologies are focussed on.

Furthermore, the comparison of steganography and cryptography is done by comparing their objectives, in other words what each technology aims to accomplish, the security services offered by each, the problems related to each technology, and also their applications.

This chapter describes measures that can be used to compare steganography with cryptography in section 2. Section 3 gives an overview of cryptography and section 4 discusses steganography in more detail. A comparison is made between cryptography and steganography in section 5.

2. COMPARISON MEASURES

First and foremost, the main difference between steganography and cryptography lies in their objectives. Cryptography focuses on keeping the contents of a message secret, while steganography focuses on keeping the existence of the message secret (Wang & Wang 2004:10). For this reason these two technologies cannot be directly compared in order to

establish which one is better. However, the comparison can be extended by comparing what services are offered by each in terms of security.

Security has traditionally been defined in terms of the three cornerstones: confidentiality, integrity, and availability (Schneier 1963:121). As a refinement of these three broad security measures, ISO 7498-2 (1989) identifies a range of five security services:

- **Identification and authentication** allow for a person to identify himself and allow the system to verify this claimed identity.
- **Authorisation** allows for the system to grant access rights as to which actions are permitted and which objects are prohibited.
- **Confidentiality** prevents an unauthorised person from reading information.
- **Integrity** prevents an unauthorised person from modifying information.
- **Non-repudiation** prevents a person from denying an action that he performed.

The comparison of steganography and cryptography can be established by examining the two technologies to determine which of the security services each one offers and how the two technologies correlate to and contrast with one another.

Furthermore the comparison should also include the problems associated with each technology and not just the security services that each technology offers. One technology that offers all five security services, but produces just as many problems, is not necessarily a better security solution than a technology that only offers one or two security services, but with no additional problems.

Finally, as part of the comparison, the application of the two technologies is compared. If steganography is used in different applications than cryptography, then steganography could not be considered to be a suitable alternative to cryptography, since the same problems are not solved.

The next section briefly discusses the main concepts that form the foundations of cryptography. The focus is on the security services offered by cryptography, as well as the common problem areas concerned.

3. CRYPTOGRAPHY

The field of cryptography has a rich and important history, ranging from pen-and-paper methods, to specially built machines, to the mathematical functions that are used today. This dissertation only briefly discusses essential information regarding cryptography.

The next section gives a definition of cryptography. The traditional uses of cryptography are discussed in section 3.2 and section 3.3 gives a brief overview of encryption algorithms and the use of a cryptographic key. Section 3.4 examines the security services offered by cryptography. Possible weakness of encryption is discussed in section 3.5.

3.1 Definition of cryptography

As defined by Gollmann (1999:200) cryptography is the science of secret writing through the enciphering and deciphering of encoded messages (Moerland 2003). It deals with the scenario where two parties, A and B, communicate over an insecure channel, with an eavesdropper possibly being able to intercept their communication as illustrated in Figure 2.1.

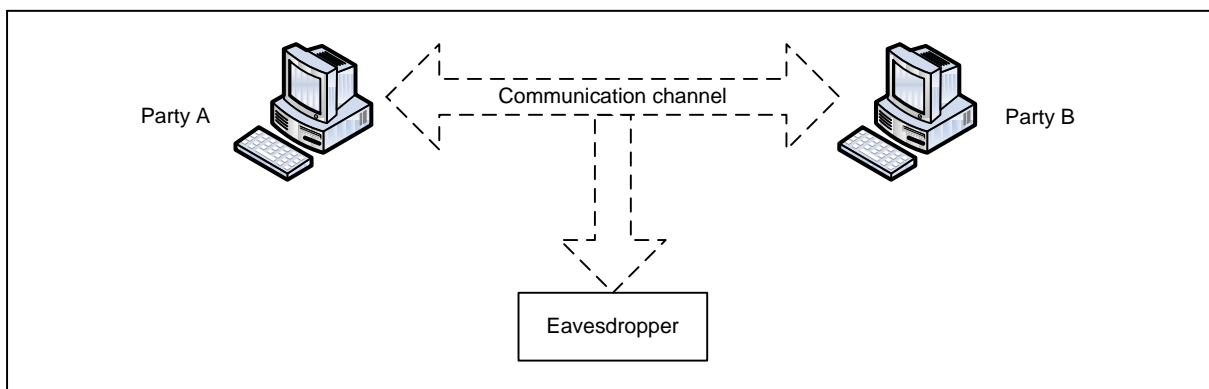


Figure 2.1. Communication over an insecure channel

Gollmann (1999:205) states that the term cryptography generally refers to a collection of cryptographic mechanisms that include:

- Encryption and decryption algorithms
- Integrity check functions
- Digital signature schemes

Encryption algorithms focus on the privacy of a secret message by scrambling the data to make it illegible to an unauthorised party. Decryption algorithms, on the other hand, unscramble the encrypted message again so that an authorised party can read it.

One example of an integrity check function is a cryptographic hash function (Whitman & Mattord 2003:13) – a mathematical function that calculates small pieces of information that can uniquely identify larger digital objects. Different objects result in different hash values. Therefore it is computationally infeasible to create another object that will have the same hash value as an existing object (Schneier 1963:94). These hash functions are used to verify that a message was not changed in transit. Another example of integrity check functions are message authentication codes (MACs) (Gollmann 1999:206). A MAC is computed from two inputs: the message and a secret cryptographic key, and also checks that information has not been tampered with.

Digital signature schemes are a mechanism for detecting whether a message was altered by an eavesdropper on the communication channel by using the same principles as asymmetric encryption (as discussed later in the chapter).

Generally, the terms encryption and cryptography are used interchangeably. Although encryption actually forms part of cryptography only the encryption part is considered in a little more detail in the following sections. All three cryptographic mechanisms are however, considered in the comparison of cryptography and steganography in section 5 of this chapter.

3.2 Traditional uses of cryptography

Primarily, the encryption part of cryptography is used as a mechanism for protecting sensitive information from unauthorised parties. This includes encrypting information for stored data, as well as encrypting information to enable secure communication (Conklin et al 2004:98). Should the eavesdropper manage to intercept a message, it should be impossible to read the message once it is encrypted.

3.3 Encryption algorithms and the cryptographic key

Auguste Kerckhoff formulated the first principles of cryptographic engineering in 1883 (Petitcolas, Anderson & Kuhn 1999:1062). The Kerckhoff principle states that the technique of encryption might be publicly known, but knowledge of the key is crucial to decrypt the message (Moerland 2003). This key is used both in the encryption phase as well as in the decryption phase, and without the key the encrypted message cannot be deciphered, even when the encryption algorithm is known.

Modern encryption algorithms can be divided into two groups, namely symmetric encryption and asymmetric encryption, based on the functionality of the keys in each technique. Also known as secret-key encryption, symmetric encryption systems require that the sender and receiver have the same secret key. This single key is required for both encryption and decryption of the message. The principle of asymmetric encryption systems, also referred to as public key encryption is that both parties, the sender as well as the receiver, have a pair of keys. One of the keys is publicly available while the other is kept private.

Both of these encryption algorithms offer security services that can counteract the vulnerabilities of communication over an insecure channel. In order to compare cryptography with steganography and ultimately reach some of the objectives stated in chapter 1, the services offered by the encryption algorithms and cryptography are examined in the next section.

3.4 Security services offered by cryptography

Confidentiality is the most fundamental security service offered by cryptography, through the implementation of encryption algorithms. Both symmetric, as well as asymmetric encryption algorithms provide the privacy of data. In both, however, it is the technique used and the length of the keys that ensure the level of secrecy of the information.

When a message is sent, both the sender and the receiver need to know that the information was not altered during the communication process. This alteration could have been intentional or unintentional. Cryptographic hash functions are thus used to ensure the integrity of data.

By using hash functions, combined with cryptographic keys, MACs provide data integrity (Gollmann 1999:205) as well as authentication. The sender uses a shared secret key to compute a MAC for a specific message. When the receiver computes the MAC and compares it with the MAC received from the sender, the receiver can determine that the message was not altered in transit. Through a comparison of the two MAC values, the receiver can also determine that the message came from the person from whom it is claimed to have come, thus offering the identification and authentication of the sender.

Digital signature schemes also offer data origin authentication (Schneier 1963), as well as support non-repudiation (Gollmann 1999:206). Based on the same principles as asymmetric encryption, digital signature schemes encrypt the message with a private key. The encrypted message acts as a signature, since only a specific private key could have produced the specific result.

To summarise, of the five security services identified by the ISO 7498-2, cryptography offers the following:

- Confidentiality
- Data integrity
- Identification and authentication
- Non-repudiation

However, the investigation into cryptography does not stop here since the problems associated with cryptography also forms part of the comparison measures. Chapter 1 mentioned some of the problems related to the use of encryption. Additional problems are discussed in the next section.

3.5 Encryption problems

Starting with the different encryption algorithms, an obvious problem with symmetric encryption is that the communication can be compromised if the key is stolen. This causes another problem: the secure distribution of keys (Schneier 1963). Key distribution involves either both parties to meet face-to-face, the use of a trusted courier, or communicating the key through an existing cryptographic channel. The first two options are often impractical as well

as unsafe, while the third depends on the security of a previous key exchange. It is also not enough to distribute the keys securely: keys have to be stored securely, used securely and ultimately destroyed securely.

Public key encryption solves the key distribution problem of symmetric encryption, but not without problems of its own. The mathematical functions that public key encryption relies on has not yet been proven to be unsolvable (Gisin et al 2002:147). At the moment algorithms to quickly calculate the mathematical relationship between the public/private key pair in order to use the one key to uncover the other, do not exist but cannot be ruled out. If a scientist were to develop such an algorithm, the encryption method might be compromised and the algorithm will be vulnerable (Gisin et al 2002:147).

Cryptography then also has the added limitations ensued by law enforcement as discussed in the first chapter.

Finally, all the security services offered by cryptography are vulnerable to cryptanalysis – the study of mathematical functions that attempts to defeat the security of cryptographic mechanisms (Menezes, van Oorschot & Vanstone 1996:15). Certain encryption algorithms, as well as certain hash functions, have already been broken by cryptanalysis (Wang & Yu 2005:1; Gilbert & Peyrin 2010:365; Bogdanov, Khovratovich & Rechberger 2011:344).

According to Gollmann (1999:207), cryptography is rarely a solution to a security problem, but more often a mechanism to convert one problem into another. By implementing cryptography in a security system, the problem is often only converted from a secure communication problem into a key management problem. This is usually done in the hope that the resulting problem will be easier to solve than the original one.

To summarise, cryptography suffers from:

- Key distribution problem
- Mathematical vulnerabilities of asymmetric encryption
- Legal limitations by governments
- Cryptanalysis

Thus far, background information on cryptography was given, security services have been discussed, as well as common problems. The question now remains whether steganography can be seen as a suitable alternative to cryptography.

In order to answer this question, the next section focuses on steganography. Basic concepts are described and the original uses of steganography are highlighted. Most importantly, the security services offered by steganography, as well as the specific problems, are examined in order to compare steganography with cryptography.

4. STEGANOGRAPHY

Although steganography is an ancient subject, the modern formulation of it is often given in terms of the *prisoner's problem* proposed by Simmons (1983:57). A more formal definition and internationally accepted terminology was agreed upon at the First International Workshop on Information Hiding (Pfitzmann 1996:347). However, the prisoner's problem is still used as a general problem statement for steganographic applications.

The prisoner's problem involves two inmates who wish to communicate in secret to hatch an escape plan. All of their communication passes through a warden who will throw them in solitary confinement should she suspect any covert communication (Chandramouli, Kharrazi & Memon 2004:35), thus they need to find a way to communicate without raising suspicion. The warden, who is free to examine all communication exchanged between the inmates, can either be passive or active. A *passive* warden simply examines the communication to try and determine if it potentially contains secret information. If the warden suspects a communication to contain hidden information, a passive warden takes note of the detected covert communication, reports this to some outside party and lets the message through without blocking it. An *active* warden, on the other hand, will try to alter the communication with the suspected hidden information deliberately, in order to try to remove the information (Anderson & Petitcolas 1998:474-81).

This section gives a brief overview of concepts used in steganography. Section 4.1 gives a definition of steganography and discusses the differences between steganography and similar technologies. The traditional uses of steganography are discussed in section 4.2. Section 4.3 discusses steganographic algorithms and the use of a steganographic key. Section 4.4

examines the security services offered by steganography and section 4.5 discusses the possible weaknesses of steganography.

4.1 Definition of steganography

Steganography is a technology concerned with ways of embedding a secret message in a cover message – also known as a cover object – in such a way that the existence of the embedded information is hidden (Anderson & Petitcolas 1998:475). A secret message can be plaintext, ciphertext, an image, or anything that can be represented as a bit stream (Johnson & Jajodia 1998(a):273). The embedding process is sometimes parameterised by a secret key, called a stego key, and without knowledge of this key it is difficult for an unauthorised party to detect and extract the secret message. Once the cover object has information embedded in it, it is called a stego object.

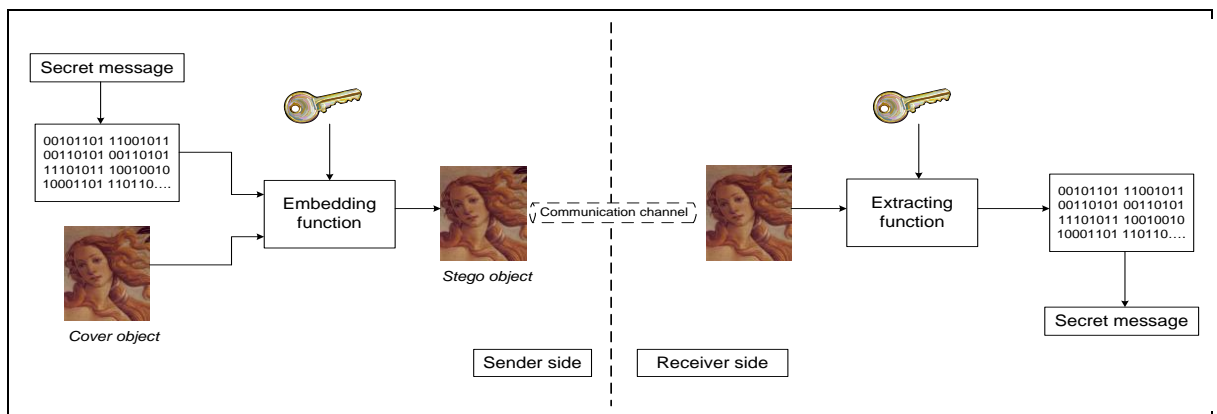


Figure 2.2. A model of the steganographic process

A general model for steganography, using an image as an example of a cover object, is illustrated in Figure 2.2. A sender embeds information in a cover object, by first applying a transform to the secret message and then manipulating a subset of the bits of the cover object to form the stego object (Anderson & Petitcolas 1998:475). The stego object is then communicated over a transmission channel, for example the Internet, to its intended recipient. At the receivers' side the process is reversed to reveal the embedded information. If a secret key was used, both the sender and the receiver should have knowledge of the key before the stego object is transmitted.

According to Katzenbeisser and Petitcolas (1999:25) a secure steganography system can be defined as a system where the original cover is indistinguishable from the stego object by both a human as well as a computer searching for statistical patterns.

By now the difference between cryptography and steganography should be evident. There are, however, other technologies closely related to steganography where the differences are not as apparent.

Two other technologies that are closely related to steganography and fall in the same domain of information hiding are watermarking and fingerprinting (Anderson & Petitcolas 1998:474-81). These technologies are mainly concerned with the protection of intellectual property. Thus, the three algorithms differ in purpose, robustness and hiding capacity (Wang & Wang 2004:10), to name but a few.

Watermarking results in all of the instances of an object to be “marked” in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection (Marvel, Boncelet & Retter 1999:1075). Fingerprinting on the other hand, embeds different, unique marks in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties (Anderson & Petitcolas 1998:476).

The most fundamental difference between the three technologies is that the object of communication for watermarking and fingerprinting is the carrier object, with the embedded data providing copyright protection (Wang & Wang 2004:10). For steganography, on the other hand, the object to be communicated is the embedded data and the carrier object serves as a disguise.

In watermarking and fingerprinting the fact that information is hidden inside the files may also be public knowledge – sometimes it may even be visible – while in steganography the imperceptibility of the information is crucial. A successful attack on a steganographic system consists of an adversary observing that there is information hidden inside a file (Artz 2001:75), while a successful attack on a watermarking or fingerprinting system would not be

to detect the mark, but to remove it. The differences between these three technologies are summarised in Table 2.1.

Table 2.1. Summary of differences between watermarking, fingerprinting and steganography

	Watermarking	Fingerprinting	Steganography
Purpose	Protect intellectual property rights	Protect intellectual property rights by identifying parties who break licensing agreements	Transmission of secret messages without raising suspicion
Perceptual invisibility	Desirable, but not crucial	Desirable, but not crucial	Crucial for embedded information not to be perceptual
Robustness against hostile removal, destruction or counterfeiting	Crucial not to be able to remove embedded information	Crucial not to be able to remove embedded information	Desirable, but not crucial
Large hiding capacity	Not important since copyright signatures are generally small	Not important since copyright signatures are generally small	Very important since it might be necessary to transmit large messages

4.2 Traditional uses of steganography

In general, steganography is used by people who wish to communicate in secret and in complete freedom. The secrecy of the communication is especially important in censored or monitored environments. Steganography can also be used to protect private communications where the use of cryptography is normally not allowed or would raise suspicion (Wang & Wang 2004:10). Alternatively, steganography can be used together with other security mechanisms to provide layered security as recommended by Conklin et al (2004:24), since if an intruder succeeds at one layer, the intruder will still need to succeed at the other levels as well.

Military and intelligence agents, especially require unobtrusive communications. Even if the content is encrypted, the detection of a signal on a modern battlefield may rapidly lead to an attack on the sender (Petitcolas, Anderson & Kuhn 1999:1063). Steganography can be employed to keep these signals hidden.

Steganography can also be used for storing information without the desire of communicating it to anyone else. Sensitive information, for example your banking details, can be embedded in a cover object that is stored on your personal computer.

4.3 Steganography algorithms and the steganographic key

There are many different steganography algorithms available, which are discussed in detail in chapter 5.

Not all steganography systems require the use of a secret key. However, the technology can be made more secure by applying the Kerckhoff principle to steganography as well. According to the principle it must be assumed that an unauthorised person has full knowledge of the design and implementation of the steganographic system. It would thus be more secure to incorporate the use of keys, either secret keys or public keys, in the implementation of steganography applications.

However, many available steganography applications still elect not to include keys in their implementations.

4.4 Security services offered by steganography

Steganography ensures the privacy of sensitive information by hiding information in other information, thus confidentiality is offered. Identification and authentication can only be offered if a steganographic key is used, since knowledge of the key can identify a person to be who he says he is. However, the manner in which the information is hidden and the techniques used could also serve as proof of identity. The technique used to embed the information thus becomes the shared secret, and when correctly embedded and extracted provides a means of identification and authentication.

The integrity of the embedded information cannot be checked, since the information could have been changed, intentionally or not, and the changes to the received information will not be noticed. A system that uses steganography to achieve data integrity is discussed in chapter 9. Since steganography do not have the functionality of authenticating the origin of

information, non-repudiation is also not offered, since someone can later deny having embedded the information.

Of the five security services defined in the ISO 7498-2, steganography thus offers confidentiality and to a lesser extent identification and authentication.

Before making a comparison at this stage, the problems concerned with steganography still need to be discussed in the next section.

4.5 Steganography problems

The biggest concern in the field of steganography is the rapid advancement of research in steganalysis, the counter-technology of steganography (Wang & Wang 2004:10). In the information hiding domain, watermarking has at first received more attention from researchers and multimedia product vendors, due to the increased importance of copyright protection. Recently though, computer specialists and security researchers have recognised that the illicit use of steganography might become a threat to the security of the worldwide information infrastructure (Kovacich & Jones 2002:35). Steganography could enable terrorists, for example, to communicate in secret without law enforcement having knowledge of this communication.

Because of this threat, researchers have actively been trying, and succeeding, to find flaws in existing steganography systems. These flaws are exploited not only for the detection of hidden information, but also include the extraction and/or destruction of the hidden data.

Steganalysis involves two major techniques: visual analysis and statistical analysis. Visual analysis tries to reveal the presence of hidden data through inspection, either with the naked eye (or ear in the case of sound) or with the assistance of a computer. Statistical analysis, on the other hand, attempts to reveal tiny alterations in a carrier objects' statistical characteristics caused by steganographic embedding (Wang & Wang 2004:10).

Both cryptography and steganography can, in essence, be misused by keeping secrets that could be harmful to innocent people. Since standardised encryption algorithms are more robust against cryptanalysis, law enforcement has instead opted for stronger regulations

regarding the use of cryptography in an attempt to reduce the communication of (potentially dangerous) information. Steganography, on the other hand, is still vulnerable to steganalysis (Wang & Wang 2004:12; Li et al. 2011:142) and the threat of a legal communication of sensitive information being intercepted and analysed, does exist.

Another possible threat to image steganography, specifically for use with secure communication, is that a firewall attached to an e-mail server could remove images from an e-mail thereby removing the secret communication. However, none of the image steganography systems proposed in this dissertation rely exclusively on e-mail as communication channel and alternative channels, for example websites, can be used to distribute stego images.

5. CRYPTOGRAPHY VERSUS STEGANOGRAPHY

To determine whether steganography can be used as an alternative to cryptography a comparison can now be made. Throughout this chapter, the objectives of the two technologies, their applications, the security services that they offer and the problems that they have, have been discussed and a summary of this information is given in Table 2.2.

Table 2.2. Comparison of cryptography and steganography

	Cryptography	Steganography
Objectives	Keeping the contents of a message secret	Keeping the existence of a message secret
Applications	Used for securing information against potential eavesdroppers	Used for securing information against potential eavesdroppers
Security services offered	<ul style="list-style-type: none"> • Confidentiality • Data Integrity • Identification and authentication • Non-repudiation 	<ul style="list-style-type: none"> • Confidentiality • Identification and authentication
Technology-specific problems	<ul style="list-style-type: none"> • Key distribution • Law enforcement • Cryptanalysis 	<ul style="list-style-type: none"> • Steganalysis • Key distribution (except with keyless steganography)

Although cryptography and steganography focus on different aspects of a message, their objectives are similar in that it entails the secrecy of the message – either its contents or its existence. Since their applications are the same, a reasonable comparison can be made.

From Table 2.2, observe that cryptography and steganography have two security services in common, namely confidentiality and identification. However, cryptography can offer two additional security services that are not offered by steganography at the moment, namely data integrity and non-repudiation.

When comparing the problems associated with steganography and cryptography it is infeasible to simply count the number of problems. Instead, impact of the problems and whether they have existing solutions should rather be considered. The biggest risk associated with steganography is the risk of steganalysis. However, as is shown in further chapters, there are techniques and mechanisms that can be applied to make steganography more robust against steganalysis. If using a secret key, key distribution can also become a problem. Again, it is shown in further chapters that the inclusion of a secret key is not crucial to the efficiency of a steganographic application, especially since the first line of defence is the fact that the information is hidden.

Cryptographic applications, on the other hand, have to include keys. However, extensive research has been dedicated to solving the key distribution problem and several solutions have been proposed (Bellare & Rogaway 1994:232; Khalili, Katz & Arbaugh 2003:342; Elboukhari, Azizi & Azizi 2010:59). Cryptography's risk of cryptanalysis is more pronounced when using proprietary software and again research has been done to protect encrypted information from cryptanalysis (Kartalopoulos 2006:146; Dajani, Owor & Okonkwo 2010:391; Szaban & Serebinski 2012:184). It seems that the only problem with cryptography that does not have a choice of several solutions is legislation. In certain countries cryptography may not be used for secure communication while steganography may still be used.

Recently, research has developed systems that combine cryptography and steganography (Bloisi & Iocchi 2007:127; Philjon 2011:217; Zhou 2011:699).

6. CONCLUSION

After an inconclusive comparison, it is still difficult to establish, with an accepted level of certainty, that steganography can be used as an alternative to cryptography. Cryptography offers more security services than steganography, but also comes with more problems. However, this does not form conclusive proof that steganography cannot be used instead of cryptography. This merely means that steganography needs to add security services to its current repertoire, while not increasing the number of problems. The goal is now to try and extend steganography so that it offers these security services as well.

It is thus still the opinion of the author that steganography does have much to offer as a security technology. This is explored in the following chapters where alternative applications of steganography are developed. The development of these applications is not done only in an attempt to extend steganography to offer more security services, but also to try and solve some of the security problems concerned with the initial classification of secure communication in chapter 1.

Throughout the dissertation, steganography is thus implemented in the following scenarios:

- Self-communication;
- One-to-one communication; and
- One-to-many communication.

However, before presenting the steganography implementations, a better understanding of steganography is gained through an intensive literature study. This understanding starts with a categorisation of the different methods of steganography in the next chapter.

CHAPTER 3

CATEGORISATION OF STEGANOGRAPHY

1. INTRODUCTION

The previous chapter briefly discussed the basic principles of steganography. There is however much more to the technology and an in-depth study of available literature should first be done in order to do research in steganography. The in-depth study begins in this chapter with a discussion of the different types of steganography.

There are mainly two approaches to dividing steganography into categories: (1) by identifying the different techniques used in the embedding process and (2) according to carrier types, i.e. the type of file used as cover object. This chapter is dedicated to these two categorisations of steganography.

Section 2 categorises steganography according to steganographic techniques and section 3 discusses a categorisation according to carrier types.

2. CATEGORISATION ACCORDING TO STEGANOGRAPHIC TECHNIQUES

There are three techniques used for embedding information in a cover object (Weiss 2009:1): insertion, substitution and generation. Data insertion techniques hide data in sections of the file that are ignored by the processing application and the technique does not modify bits that are relevant to the end user.

Substitution-based techniques replace data from the cover medium with data from the secret message. This does not result in a larger cover file; however, depending on the cover medium and steganographic algorithm used, substitution may result in degrading the cover object (Fridrich 2010:55).

Generation techniques create a cover object specifically for the purpose of hiding the secret message. The properties of the generated cover object are usually dependent on the secret message structure (Fridrich 2010:55). While insertion and substitution techniques can be

discovered by comparing the stego object with the original object, generation techniques are immune to comparison tests since the result of a generation algorithm is the original object.

Kipper (2003:39) identified a further six categories, namely substitution, transform domain, spread spectrum, statistical method, distortion and cover generation techniques. These six categories, can also fall within the three broader categories of steganography techniques. Kipper's six techniques can be merged with the original three categories resulting in Table 3.1.

Table 3.1. Steganography technique categories

Technique	General categorisation	Explanation
Substitution system techniques	Substitution	Redundant bits from the cover object are replaced with bits from the secret message
Transform domain techniques	Substitution	Changes made to the cover object during compression, are used to hide information
Spread spectrum techniques	Substitution	The secret message is embedded in noise and then combined with the cover object
Statistical method techniques	Substitution	Only one bit is embedded in the cover object resulting in a statistical change
Distortion techniques	Insertion	A change in the cover object is created to hide information that can be recovered when comparing the changed object with the original
Cover generation techniques	Generation	A cover object is created for the purpose of hiding information

As illustrated in the table, substitution is the most popular technique. Substitution techniques do not add information to the cover object and thus do not increase the size of the object – a process that is easily detectable. However, the disadvantage of substitution is that the amount of data of the original object to replace needs to be carefully selected. If not carefully selected, the changes might become perceivable to someone looking for hidden information. Most steganographic algorithms implement substitution techniques. This dissertation thus focuses on substitution techniques in the discussion of image steganography algorithms done in chapter 5 since substitution techniques are the most studied steganography techniques today (Fridrich 2010:53).

Categorising steganography based on the techniques used, is one approach. An alternative approach is to categorise steganography based on the types of digital files that are used as carriers for the embedded information. This approach to categorisation is examined next.

3. CATEGORISATION ACCORDING TO CARRIER TYPES

In digital files, redundant bits are defined as the bits of an object that provide file quality far greater than necessary for the object's use and rendering (Currie & Irvine 1996:194), for example image files that can display 16-million different colours, while the human eye is only able to perceive about 10-million different colours (Owens 2002:9). The redundant bits of an object are those bits that can thus be altered without the alteration being detected easily (Anderson & Petitcolas 1998:474). In steganography, file formats with a high degree of redundancy is preferable since redundant bits can be replaced with secret information without the embedded information being perceivable.

Image and audio files especially comply with this requirement of redundancy, while research has also uncovered other file formats that can be used for information hiding. Figure 3.1 shows the four main categories of file formats that can be used for steganography.

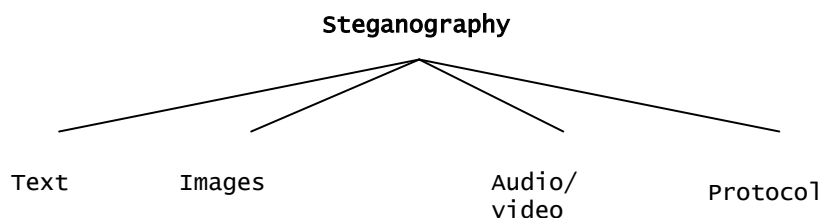


Figure 3.1. Categories of steganography based on carrier types

Each of these file format categories uses different techniques for hiding information based on the unique characteristics of the file format and the redundancy created in the digital representation of the file. Text steganography is briefly discussed in section 3.1, image steganography in section 3.2, audio/video steganography in section 3.3 and protocol steganography in section 3.4.

3.1 Text Steganography

Hiding information in text is historically the most important method of steganography. One method that is used to hide a secret message in text is called a null cipher where every n^{th} letter of every word of a text message is used to hide a letter of the secret message (Rabah 2004:245).

Another text steganography technique is known as a book cipher (Anderson & Petitcolas 1998:474). A publicly available source, for example a book or a newspaper, is used as a cover object. A code that consists of a series of pointers to characters, is shared among the involved parties. For example, the cipher group “54316” might mean page 54, line number 3, the 16th character. Discovering the secret message relies solely on gaining knowledge of the secret code (Krenn 2004:3).

It is only since the beginning of the Internet and all the different digital file formats that null ciphers and book ciphers have decreased in importance (Moerland 2003).

In the digital world, small modifications to font size, font style, line spacing, boldness and other text formatting procedures can be applied for steganography. Existing text steganography programs use additional white spacing or tabbing at the end of a line. In this way a tab at the end of a line might indicate a one and the absence of a tab might indicate a zero (Moerland 2003).

Although a number of different techniques can be defined for hiding information in text, (Shirali-Shahreza 2008:1912; Por, Ang & Delina 2008:735) text steganography using digital files has decreased in popularity since text files have a very small amount of redundant data.

3.2 Image Steganography

Due to the large amount of redundancy created in the manner in which digital images are represented, images are the most appropriate carrier type for steganography. Steganography on images is also the most popular form of steganography, since images occur frequently on websites, as e-mail attachments, etc. There is thus minimum cause for suspicion when a digital image is used.

Given that images are ideal carriers, as well as popular information media, this dissertation focuses only on image steganography in subsequent chapters and applications.

3.3 Audio/Video Steganography

Audio compression is mainly based on research that has been done on the biological properties of the human ear, specifically on the amount of data that can be removed from the audio file without the removal being audible (Bandyopadhyay et al 2008:109). Audio compression algorithms, for example MPEG Model 1 Layer III (MP3), exploit these properties in order to obtain small file sizes without losing sound quality (Atoum et al 2011:184). These properties can also be used for audio steganography by hiding information in audio files without the difference being audible. The digital representation of audio includes representing the sound intensity at a certain point in time. Since a 16 bit audio file typically has 2^{16} levels for this sound intensity, a difference of 1 level will be unnoticeable by the human ear.

A technique that is unique to audio steganography is masking, where a faint, but audible, sound becomes inaudible in the presence of another louder audible sound (Kipper 2003:53). Echo hiding is another technique where an inaudible echo is added to an audio file (Bender et al 1996:332).

Although nearly equal to images in steganographic potential, the larger size of meaningful audio files makes them less popular to use than images (Artz 2001:75).

In general, video files can be seen as a collection of images and sounds, thus most image and sound steganographic techniques can be used on video as well (Papapanagioutou et al 2005:589). Added advantages of video steganography are that videos can conceal a large amount of data. The fact that it is a moving stream of images and sound is also beneficial, since otherwise noticeable distortions will not be picked up so easily by humans. A disadvantage of video steganography is the large size of a video clip that is not regularly communicated over normal transmission channels.

3.4 Protocol Steganography

The term protocol steganography refers to the technique of embedding information within the volatile data created in network transmissions (Rabah 2004:250).

A network packet consists of packet headers, user data, and packet trailers. All the packets sent across a network following the OSI network model, have the same packet structure. Covert channels where steganography can be used exist in the layers of the OSI network model (Handel & Sandford 1996:23). Information can be hidden in redundant parts of messages and network control protocols can be used to transmit packets over the network.

Ahsan and Kundur (2002) provide one such example of where information can be hidden in the header of a TCP/IP packet. Fields that are either optional or are never used are ideal for hiding information. Each TCP packet segment begins with a uniformly formatted 20-byte header of which 6 bits are not utilised by the protocol (Rabah 2004:251). All these bits could be used to store the secret message.

4. CONCLUSION

This chapter divided steganography into categories to show the different types of steganography methods that exist. Of the different carrier types that are suitable for steganography, digital images are the most common type of file for which steganographic applications are currently available (Fridrich 2010:xvii). The remainder of this dissertation thus focuses only on image steganography. The next chapter discusses different image file formats and the compression techniques that are used on them. A discussion on image file formats and compression is necessary before the algorithms for image steganography can be examined.

CHAPTER 4

DIGITAL IMAGES AND COMPRESSION

1. INTRODUCTION

The understanding of image steganography requires a fair amount of background information regarding the presentation and properties of digital images. This chapter gives an overview of digital image concepts, such as colour representation, how images are stored, and the structure of a digital image. Image definition, image compression, and the different kinds of image file formats are also discussed in this chapter to serve as background information to understanding image steganography, since image steganography exploits image features as a mechanism for hiding information.

The concepts covered in this chapter were chosen for its relevance to applications in image steganography and is therefore not a concise list of image definitions, image compression methods or image file formats.

The next section discusses digital imaging concepts such as colour representation and image definition is discussed. Section 3 provides details on image compression techniques and section 4 examines typical image file formats in the categories of spatial domain and transforms domain file formats.

2. DIGITAL IMAGING CONCEPTS

In order to fully understand how information is embedded in images there are a few concepts in the field of digital imaging to consider. Colour representation is discussed in section 2.1 and image definition is discussed in section 2.2.

2.1 Colour representation

Visible light is comprised of electromagnetic waves and colours are described by the amount of energy present at a specific wavelength (Fridrich 2010:15). The human eye is capable of distinguishing only a relatively small number of possible colours, although uncountable many

colours exist (Fridrich 2010:15). According to the trichromatic theory of colour (Boothe 2002:199) each colour that the human eye can perceive can be obtained from three basic colours: red, green and blue. This theory is used in the additive colour model (Fridrich 2010:16) where every digitally represented colour is represented as a linear combination of red, green and blue components. The additive colour model is also known as the RGB colour model, with the amount of each colour denoted by R, G and B (Foley et al 1994).

Another popular colour model is the YUV colour model or luminance/chrominance model – with luminance Y defined as a weighted linear combination of the RGB channels, while chrominances U and V convey colour information (Sattarova & Tai-hoon 2009:44). When transformed so that Y, U and V are represented by 8-bit integers, the colour model is known as the YC_rC_b colour model (Fridrich 2010:17).

2.2 Image definition

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image (Johnson & Jajodia 1998(b):26). Individual points are referred to as pixels and the pixels form a rectangular map of where each pixel is located and its colour (Murray & van Ryper 1996:124).

The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel (Owens 2002:8). For example, if an image's bit depth is 8, then 8 bits are used to describe the colour of each pixel and a total of 256 different colours can be displayed. Figure 4.1 shows an example of a greyscale image with bit depth 8 that can display 256 different intensities of grey.

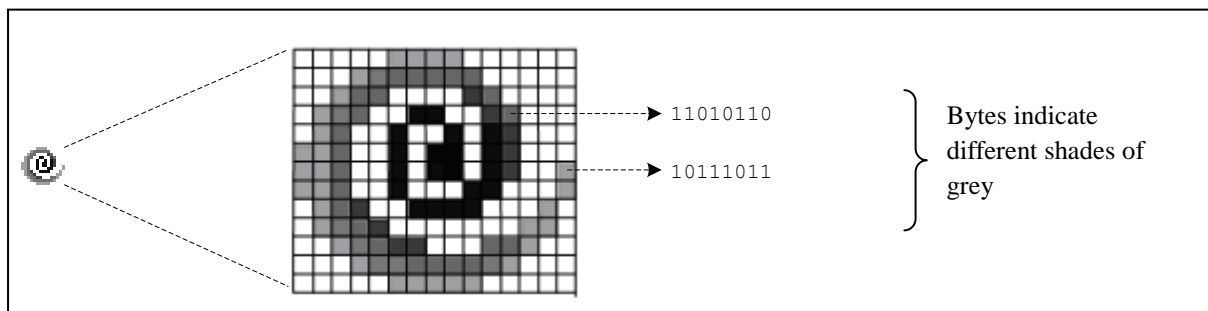


Figure 4.1. Pixels and bit representation of a greyscale image with bit depth 8

Digital colour images are typically represented with bit depth 24 and use the RGB colour model, also known as true colour (Schneider & Gersting 2004:146). All colour variations for the pixels of a 24-bit image are derived from the three primary colours: red, green and blue, and each primary colour is represented by 8 bits (Johnson & Jajodia 1998(b):26). Figure 4.2 illustrates the use of 3 bytes per pixel in 24-bit images.

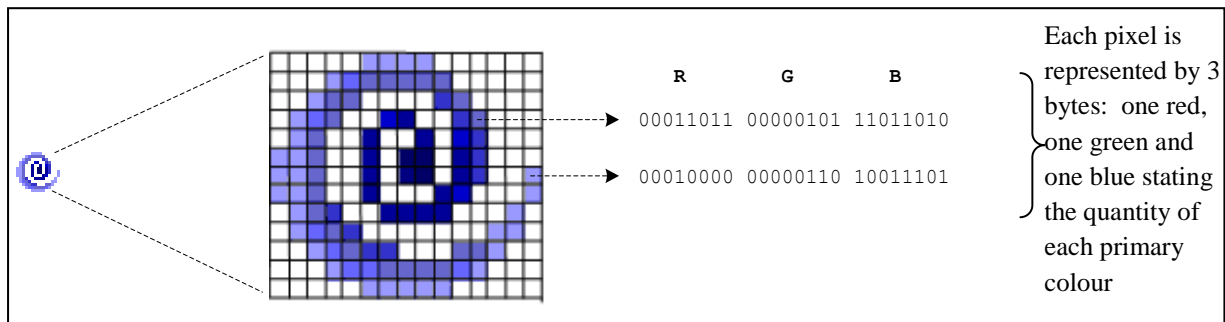


Figure 4.2. Pixels and bit representation of a 24-bit colour image using the RGB colour model

In one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colours. For comparison, a quality offset printing press can print about 4000 colours, a traditional film photograph can contain in the region of 6-million colours and the human eye can recognise approximately 10-million colours (Owens 2002:9). Evidently, a large amount of redundancy is created that a steganography algorithm can utilize to its advantage.

Image compression techniques are also important in understanding how information can be embedded in digital images and are discussed next.

3. IMAGE COMPRESSION

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard network connection. In order to display an image in a reasonable amount of time and use a reasonable amount of space to store the image, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyse and condense image data, resulting in smaller file sizes. This process is called compression (Scheider & Gersting 2004:147).

Two types of image compression methods exist: lossy and lossless (Moerland 2003:4). Both methods save storage space, but the procedures that they implement differ. The subsections

that follow discuss the difference between lossy and lossless. Lossless compression is discussed in section 3.1 and lossy compression is discussed in section 3.2. The relationship between compression and steganography is discussed in section 3.3. An example of each of the compression methods is given later in the chapter when discussing different image formats.

3.1 Lossless compression

Lossless compression represents data in mathematical formulas while not removing any information from the original image. The original image's integrity is maintained and the decompressed image output is bit-by-bit identical to the original image input (Schneider & Gersting 2004:149).

3.2 Lossy compression

Lossy compression, on the other hand, creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate, resulting in close approximations of the original image, although not an exact duplicate (Schneider & Gersting 2004:149).

3.3 Compression and steganography

Compression plays a very important role in the design of steganographic algorithms. Lossy compression techniques result in smaller image file sizes, but increases the possibility that the embedded message may be partly lost due to the fact that redundant image data is removed (Dunbar 2002:5). Lossless compression keeps the original digital image intact without loss of image detail. However, the image is not compressed to such a small file size (Johnson & Jajodia 1998(b):32). Different steganographic algorithms have been developed for both of these compression types. The image steganography algorithms are discussed in the next chapter.

4. IMAGE FILE FORMATS

The way that images are stored differs mostly in the digital representation of the image and the level of compression. The image file format usually depends on the intended use of the image, since different image file formats were developed with a specific purpose in mind. Although there are a large variety of image file formats available, the ones explained in the following sections are regarded as relevant to image steganography and most image file formats can be seen as a variation of one of these formats.

Image formats can be divided into two domains: spatial domain and transform domain. An image in the spatial domain format is represented as a dense rectangular grid of pixels (Fridrich 2010:18). The human visual system, however, does not perceive an image as a grid, but rather perceives an image as a collection of segments filled with texture. An image in the transform domain format is thus represented as mathematical formulas based on compression techniques to allow for a higher rate of compression (Fridrich 2010:22).

It is important to appreciate the different image file formats that are available since the method of embedding information with image steganography is different for each file format. Section 4.1 gives an overview of spatial domain formats and section 4.2 discusses transform domain formats.

4.1 Spatial domain formats

Spatial domain formats can be divided into raster image formats and palette based image formats. The next section discusses raster images and palette based images are discussed in more detail in section 4.1.2. Each section gives image file format examples of the images and discusses the compression methods used for each format, if applicable.

4.1.1 Raster images

In a raster image format, an image is represented in a row-by-row grid of pixels with one or more bytes used to store one pixel depending on the bit depth (Fridrich 2010:18). Figure 4.1 and 4.2 are examples of how raster images are stored with different bit depths. A Microsoft

Windows bitmap file (BMP) is an example of an image file format that stores the information as a raster image.

The BMP format is one of the simplest image file formats in use. Images are stored with a bit depth of 1 (2 colours), 4 (16 colours), 8 (256 colours), 16 (65 536 colours) or 24 (16.7 million colours). In the BMP image file format compression is optional, but lossless compression can be used if compression is required (Murray & van Ryper 1996:125).

A lossless technique that can be used with BMP file formats is run length encoding (RLE) (Salomon 2004:20). This method of compression replaces a sequence of identical values, v_1, v_2, \dots, v_n , with a pair of values (v, n) which indicates that the value v is replicated n times (Schneider & Gersting 2004:147).

Two methods exist for run length encoding: The first method compresses an image by finding duplicate adjacent pixels, for example pixels of which the red, green and blue components are the same for images of bit depth 24. These pixels are compressed into pixel pairs that state the number of times that the specific pixel value is replicated. For example, the following grid of identical pixels in a 24-bit colour image:

Red	Green	Blue
(10100110	11000100	00001100)
(10100110	11000100	00001100)
(10100110	11000100	00001100)

can be compressed to 3 10100110 11000100 00001100 meaning that the pixels are duplicated three times. The higher the frequency of a specific colour, for example in an image with a solid-colour block, the higher the compression rate.

The second method for run length encoding is to compress each colour separately. Adjacent pixels with the same value for a specific colour component can be compressed regardless of the values of the other two colour components. This approach does not rely on large areas of the same colour, but rather on the repetition of the same intensity of a specific colour.

BMP images may result in very large files (Fridrich 2010:18), but remains a popular image file format because of its simplicity. BMP files are also popular for image steganography,

because it has the capacity to hide relatively large messages (Fridrich 2010:18). Image steganography algorithms that use BMP files are discussed in chapter 5.

4.1.2 Palette based images

Palette based images are another popular image file format commonly used on the Internet (Tzeng, Yang & Tsai 2004:791). Images such as computer-generated graphics, line drawings and cartoons are often stored using palette based images (Fridrich 2010:19). The most widely known palette based image file format is GIF (Graphical Interchange Format) (Johnson & Jajodia 1998(b):26). The format specifications of a GIF image define that a GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that can be used to colourise a GIF is 256 (Wiggins et al 2001:789).

GIF images are indexed images where the colours used in the image are stored in a palette, sometimes referred to as a colour lookup table (Wong, Cheung & Po 2002:949). Each colour in the palette is stored as an 8-bit RGB colour. Every palette based image consists of two parts: the palette and the image data. The image data consists of a rectangular grid of 8-bit indexes that point to the palette (Johnson, Duric & Jajodia 2001:16). The pixels, thus, do not store the colours themselves. Figure 4.3 demonstrates the use of the colour palette.

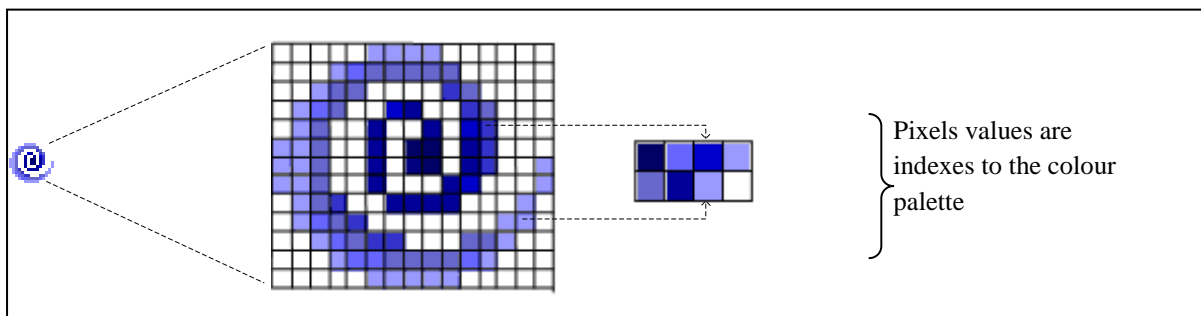


Figure 4.3. Pixels and indexes of an 8-bit GIF image with image palette

The palette based image format is a lossless format. However, when converting a raster image to a palette based image, a loss of detail can occur. To convert a 24-bit RGB raster image to a palette based image involves creating the colour palette and mapping the original colours to the newly created palette (Fridrich 2010:19). If the original image contained more than 256 distinct colours, the number of colours in the original image needs to be reduced. Colour quantization is a process used to reduce the number of original colours to fewer

distinct colours to fit on the palette with the least possible visual distortion (Orchard & Bouman 1991:2678). Colour quantization is a lossy process.

Once the palette has been obtained, the original colours are mapped to the colour palette through a process called dithering (Fridrich 2010:19). For original colours that are not in the colour palette, an approximation is found that results in the least visual distortion. Dithering is also a lossy process.

There are many different algorithms for both colour quantization and dithering (Fridrich 2010:19). However, these techniques are outside of the scope of this dissertation.

4.2 Transform domain formats

Transform domain techniques focus on representing images that are easy to compress (Fridrich 2010:22). Such techniques are normally lossy and thus form an approximation of the original image with some loss of detail. An example of an image format that makes use of the lossy compression technique is the Joint Photographic Experts Group (JPEG) (Johnson & Jajodia 1998(b):28) file format. The JPEG file format is the most popular image file format on the Internet, because of the small size of the images. It is especially good at compressing photographic images of real world scenes or objects and is commonly used by software for digital cameras and scanning devices (Fridrich, Goljan & Du 2001:276).

JPEG compression makes use of the discrete cosine transform (DCT) to transform the image into an easily compressible form (Fridrich 2010:22).

To compress a JPEG image, the RGB colour representation is first converted to an $Y C_r C_b$ representation (Currie & Irvine 1996:196). The human eye is more sensitive to changes in the brightness of a pixel than to small changes in colour (Fridrich 2010:22; Currie & Irvine 1996:196). This fact is exploited by the lossy compression scheme by downsampling the chrominance component to reduce the size of the file. This component of lossy compression is illustrated in Figure 4.4.

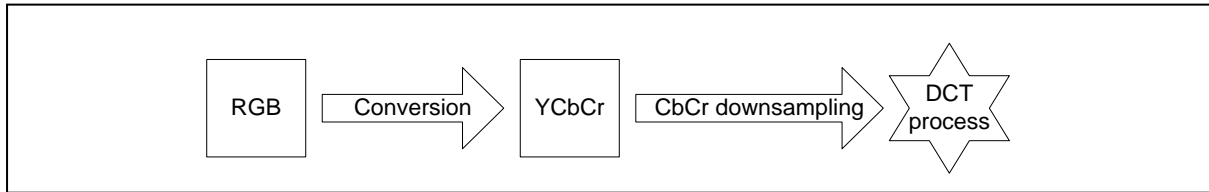


Figure 4.4. RGB conversion and UV downsampling

The next step is the actual transformation of the image. For JPEG images, the DCT is used, but similar transforms are for example the discrete fourier transform (DFT) and discrete wavelet transform (DWT) (Naghsh-Nilchi & Pourmohammadbagher 2006:147).

The DCT transforms a signal from the spatial domain into a frequency representation - the transform domain. The pixels are first grouped into 8×8 pixel blocks. Each pixel block is then transformed into 64 DCT coefficients (Fridrich, Goljan & Du 2001:276) using the DCT mathematical formula.

The compression now relies on two techniques to reduce the data required to store an image:

1. Quantization of the image's DCT coefficients to reduce the number of possible values of a quantity, thereby reducing the number of bits needed to represent the image (Fridrich 2010:25).
2. Entropy coding of the quantized coefficients to represent the quantized data as compactly as possible (Kipper 2003:50).

During quantization the DCT coefficients are first divided by an integer value (Fridrich 2010:23). The integers used in the division are referred to as quantization steps and their values are recommended by the JPEG standard. Quantization steps are larger for higher frequencies, thus making sure that high frequencies become very small (Moerland 2003). Larger quantization steps produce smaller file sizes through higher compression, but introduce more visual distortion (Fridrich 2010:25).

After the division the results are rounded to integer values (Currie & Irvine 1996:197) – the lossy part of the algorithm. For high frequencies this will mostly be zero, resulting in large sequences of zeros which are easier to compress (Moerland 2003). The coefficients are then encoded using entropy coding, for example Huffman coding (Fridrich 2010:23), to change

the colour frequencies into numeric values and further reduce the size (Kipper 2003:50). The DCT process is depicted in Figure 4.5.

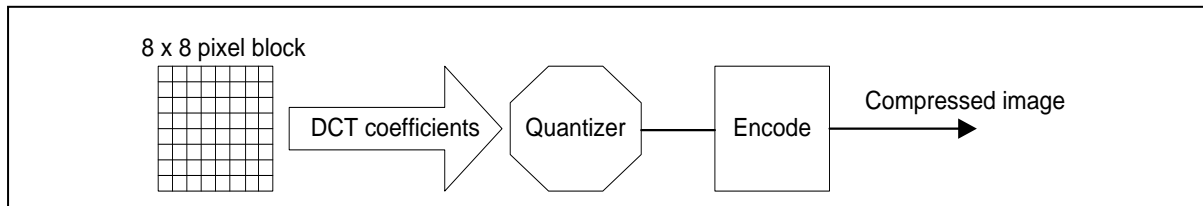


Figure 4.5. The discrete cosine transform (DCT) process

5. CONCLUSION

This chapter discussed the format and compression of image files that were deemed important for the continuation of a study in image steganography. Image formats in the spatial domain and in the transform domain were discussed. The relevant compression methods in each domain were also discussed. The next chapter examines image steganography in the spatial and transform domain.

CHAPTER 5

IMAGE STEGANOGRAPHY

1. INTRODUCTION

This chapter focuses on image steganography and provides necessary background information for the rest of the dissertation. The ultimate objective is to apply image steganography techniques to different secure communication categories to determine if image steganography complies with the requirements of secure communication. To reach this goal, suitable algorithms for different applications should be identified and therefore the technical details of the algorithms are examined in this chapter.

Chapter 3 divided steganography techniques into substitution, insertion and generation techniques. This chapter focuses on substitution techniques since it is the most practical approach for communicating large amounts of information and the mainstream approach to image steganography (Fridrich 2010:49).

Raster and palette based images of the spatial domain and JPEG images of the transform domain, as discussed in chapter 4, are popular image file formats. Therefore, this chapter only examines steganography algorithms that were specifically developed for images in the spatial domain and transform domain. Steganography algorithms that are based on information theory, statistical physics or signal processing were thus excluded from this chapter.

This chapter, however, first discusses evaluation criteria for image steganography algorithms. To evaluate whether an image steganography algorithm is suitable for a specific application, algorithms are evaluated based on a set of criteria. Evaluation criteria are discussed in the next section and are referred to in the subsequent discussions of image steganography algorithms later on in the chapter.

Spatial steganography algorithms are discussed in section 3 and transform domain steganography in section 4. Since the data extraction process is usually the inverse of the

embedding process, only the data embedding process for each steganography algorithm is discussed.

Finally section 5 gives a summary of how the image steganography algorithms discussed in sections 3 and 4 comply with the evaluation criteria discussed in section 2.

2. EVALUATION CRITERIA

Wang and Wang (2004:78) identified invisibility, payload capacity, and robustness against image manipulation attacks as three important requirements of an image steganography algorithm. Fridrich (2010:13) added statistical undetectability as another important requirement.

The evaluation criteria are described below:

- **Invisibility** – The invisibility of the embedded information is the first and foremost requirement, since the strength of image steganography lies in its ability to be unnoticed by the human eye. The moment that tampering of an image becomes noticeable, the algorithm is compromised.
- **Payload capacity** – Payload capacity is the amount of information that can be embedded in a digital image without visible image distortion. Since image steganography is used for hidden communication, algorithms should be able to accommodate sufficiently large hidden messages.
- **Robustness against image manipulation attacks** – During communication of a stego image between authorised parties, the image may undergo changes by an active warden in an attempt to remove hidden information. It is thus important for steganographic algorithms to be robust against malicious as well as unintentional changes to the image.
- **Statistical undetectability** – Many steganographic algorithms leave a signature when embedding information that can easily be detected through statistical analysis. For an

algorithm to be statistically undetectable, it should be impossible for a warden to statistically prove the existence of hidden information.

The degree to which the image steganography algorithms comply with the above criteria is included in the discussion of each of the algorithms in the next two sections.

3. SPATIAL DOMAIN STEGANOGRAPHY

Spatial domain steganography uses images in the spatial domain format for hiding information. Spatial domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation to embed information (Johnson & Jajodia 1998(a):273). Data embedding is done by directly replacing data of the image pixel values with secret information (Li et al 2011:146). Spatial domain steganography algorithms take advantage of the large amount of redundant data that is created in the way that digital images are stored in the spatial domain (Kipper 2003:41).

Image steganography algorithms that can be applied to raster images are discussed in section 3.1 and section 3.2 discusses image steganography algorithms for palette based images.

3.1 Raster images

The best-known algorithm developed for raster images is the least significant bit (LSB) algorithm (Fridrich 2010:59). The last bit of a byte is considered the least significant bit, since changes in its value have the least effect on the information that the byte is representing. Section 3.1.1 gives a brief overview of LSB embedding and its strong points. The algorithm's weaknesses are then analysed in section 3.1.2 and section 3.1.3 discusses an algorithm that builds on LSB embedding but provide more security.

3.1.1 Overview of LSB embedding

Least significant bit (LSB) embedding makes use of the small differences created when changing the least significant bit of a byte and is a common, simple approach to embedding information in a cover image (Johnson & Jajodia 1998(b):28). The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the

secret message. A popular approach is to embed bits from the secret message along a pseudo random path generated from a stego key shared by the sender and the receiver – such a path is called the selection channel (Fridrich 2010:54,60).

When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. The payload capacity can thus be as high as three times the number of pixels in the image (Moerland 2003). In spatial domain steganography, the embedding rate of an algorithm is described as bits per pixel (bpp) (Li et al 2011:146). LSB embedding in a 24-bit colour image, thus has an embedding rate of 3 bpp.

Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes to the intensity of the colours. These changes cannot be perceived by the human eye (Fridrich 2010:60), thus LSB embedding complies with the requirement of invisibility. With a well-chosen image, the message can even be hidden in the least as well as second to least significant bit while maintaining invisibility (Johnson & Jajodia 1998(b):28).

LSB embedding can be divided into two broad categories: fixed-size insertion methods and variable-size insertion methods (Potdar, Han & Chang 2005:717), depending on the number of LSBs of each byte used for embedding. Fixed-size insertion methods use a fixed number of LSBs to embed the secret in each byte of the cover image (Lou & Liu 2002:449). Variable-size insertion methods use a variable number of LSBs from each byte of the cover image to embed information according to each pixel's suitability for embedding (Lou & Liu 2002:449). Pixels in large fields of monochrome colour or pixels that lie on sharply defined boundaries should be avoided during embedding, since changes to these pixels could result in visual distortion of the image. However, pixels in areas of high contrast and high luminance can accommodate more bits from the secret message without a noticeable difference (Lou & Liu 2002:449). Algorithms that are designed to avoid embedding in unsuitable areas of an image, are called adaptive steganography algorithms (Fridrich 2010:54).

3.1.2 Weaknesses of LSB embedding

LSB embedding is a popular image steganography algorithm due to its ease of implementation (Krenn 2004:4). There are, however, weaknesses to the algorithm, mainly

that the embedded information is easy to detect should an attacker be aware of the technique used (Wang & Wang 2004:10).

LSB embedding is also sensitive to image manipulation attacks, such as image cropping, resizing, colour space conversion or resampling (Venkatraman, Abraham & Paprzycki 2004:347). The LSB algorithm is not suitable for images compressed with lossy compression since the least significant bit is often seen as redundant by lossy compression algorithms and is thus removed during compression. LSB embedding is thus not robust against image manipulation attacks.

LSB embedding is also not resistant to statistical attacks, since the embedded information is easy to detect. A fairly simple statistical attack, called a histogram attack, can be used to detect the presence of embedded information by studying the histogram of a stego image (Fridrich 2010:63). The process of changing the LSBs of the cover image during LSB embedding, leads to characteristic artefacts in the image histogram that can be used to identify that steganography has been used (Xi, Ping & Zhang 2010:203).

3.1.3 Improvements to LSB embedding

Changing LSB values for LSB embedding results in an unnatural histogram that is easy to detect (Fridrich, Soukal & Goljan 2005:596). A trivial modification to LSB embedding that improves on LSB embedding's vulnerability to statistical detection is an algorithm called ± 1 embedding (Li et al 2011:147). In the embedding phase of the ± 1 embedding algorithm, when an LSB needs to change, instead of flipping the bit to the opposite bit value, the value stored in the byte is increased or decreased by one (Fridrich 2010:119). This has the effect of modifying the LSB, but may modify other bits as well. For example, if the original byte stored the value 127 (01111111_2) and it is increased by one it changes to 128 (10000000_2). The ± 1 embedding algorithm is harder to detect than LSB embedding since it does not leave a clear signature on the histogram of the stego image (Fridrich, Soukal & Goljan 2005:596).

3.2 Palette based images

In palette based images, an image is stored as pointers to colours on a palette. LSB embedding can be used to hide information in a palette based image with a few adjustments

to the original LSB algorithm. Section 3.2.1 gives an overview of LSB embedding in palette based images and the adjustments that should be made to LSB to accommodate palette based images more efficiently. Section 3.2.2 discusses the weaknesses of LSB in palette based images. Optimal parity embedding, which is an advanced steganography algorithm for palette based images is discussed in section 3.2.3.

3.2.1 LSB embedding in palette based images

Since the pixels of a palette based image store indices and not colours, changes to the LSBs of pixels could result in visual distortion. Should the least significant bit of a pixel be changed, the pixel could display a completely different colour since the index to the colour palette has changed (Johnson & Jajodia 1998(c):113). If adjacent palette entries are similar, there might be little or no noticeable change, but should the adjacent palette entries be very dissimilar, the change would be evident and the hidden information would be visible. Changes between colour values of adjacent palette entries may change gradually but rarely, if ever, in one bit shifts (Johnson & Jajodia 1998(a):273).

A simple solution to avoid drastic colour changes is to preprocess the palette (Fridrich 2010:69). One approach to preprocessing the palette is to sort the palette so that the colour differences between consecutive colours are minimized (Wang & Wang 2004:79).

Another approach is to decrease the number of colours in the palette before embedding. Once the number of distinct colours has been decreased, colours are again added to the colour palette that are close to the original colours, but with a different index (Katzenbeisser & Petitcolas 1999:53). If, for example, the original palette consisted of the maximum 256 colours, the colours are first decreased through colour quantization to 128. For each of the 128 colours, a new colour is added that is identical to the original colour, but with a different index. If the palette is then sorted, changes to the LSB of an index would point to a copy of the original colour. If the palette is decreased to 128 colours, the embedding rate is 1 bpp. However if the original colours are decreased to 64 or 32, an embedding rate of two or three bits per pixel can be achieved (Fridrich 2010:69).

3.2.2 *Weaknesses of LSB embedding in palette based images*

The main weakness of LSB embedding in palette based images is the nature of the palette based image itself and the possibility of visual distortion should the indices be changed.

Preprocessing the palette increases the invisibility of the embedded information, but also increases the detectability of the embedded information since tampering with the palette leaves a clear signature. A palette sorted according to colour values is unlikely to occur naturally (Fridrich 2010:70), and a palette with groups of two, four or eight identical colours is suspicious as well.

The optimal parity embedding algorithm was developed by Fridrich and Du (1999:47-60) to hide information in palette based images without changes to the palette.

3.2.3 *Optimal parity embedding*

Optimal parity embedding assigns each colour in the colour palette a parity bit (0 or 1) based on that colours' red, green and blue values (Fridrich & Du 1999:50). The parity bit P is calculated as

$$P = (R + G + B) \text{ mod } 2$$

When embedding a secret message, a pixel is selected for each bit of the message and a comparison is made between the pixel's parity bit and the message bit. If they are not the same, the algorithm determines the closest colour in the palette with the opposite parity. When this colour is found, the index of the pixel is changed to point to the closest colour.

Information is thus not hidden in the LSB values of the pixels, but in the parity bits of the pixels. The visual distortion to the image is kept to a minimum since pixels are altered to point to similar colours in the palette. An embedding rate of 1 bpp can be achieved (Fridrich 2010:73) and since the palette is not tampered with, the embedded information is much harder to detect than if the palette was preprocessed (Fridrich 2010:71).

Additional spatial domain steganography algorithms such as bit-plane complexity segmentation (BPCS) steganography (Kawaguchi & Eason 1999:464), pixel value

differencing (PVD) steganography (Wu & Tsai 2003:1613), multiple base notational system (MBNS) steganography (Zhang & Wang 2005:67) and others exist, but is outside the scope of this chapter since the function of this chapter is to merely serve as an introduction to image steganography.

4. TRANSFORM DOMAIN STEGANOGRAPHY

Steganography in the transform domain involves the manipulation of image transforms (Johnson & Jajodia 1998(a):273). These techniques hide messages in more significant areas of the cover image, making it more robust against image manipulation attacks (Katzenbeisser & Petitcolas 1999:56) and the embedded message can survive conversion between lossy and lossless compression (Provos & Honeyman 2001)).

The following subsections use JPEG steganography as an example of transform domain steganography. Section 4.1 gives an overview of JPEG steganography followed by a discussion of the algorithm's weaknesses in section 4.2. More advanced steganography algorithms for JPEG image, namely Outguess and F5, are then briefly discussed in sections 4.3 and 4.4.

4.1 JPEG steganography

The JPEG compression algorithm is divided into lossy and lossless stages. The DCT and the quantization phase defined in the previous chapter, form part of the lossy stage, while the Huffman encoding (Fridrich 2010:23) used to further compress the data is lossless. Since steganography usually hides information in redundant data and redundant data is removed during lossy compression, steganography cannot take place during the DCT and quantization phases. Steganography can, however, take place between the lossy and lossless stages. For JPEG steganography, LSB embedding is used to embed the message into the least significant bits of all non-zero DCT coefficients before applying the Huffman encoding (Kipper 2003:50). Instead of embedding the information in the pixels, in other words in the spatial domain, the information is embedded in the DCT coefficients, in the transform domain, making the hidden information invisible (Fridrich 2010:67).

The embedding rate for transform domain steganography algorithms is measured in bits per non-zero coefficient (bpnc) (Li et al 2011:150). The embedding rate for JPEG steganography is thus 1 bpnc (Fridrich 2010:67).

4.2 Weaknesses of JPEG steganography

The histogram of the DCT coefficients of a natural image, shows a definite symmetrical distribution of coefficients with a spike around zero for all JPEG images (Fridrich 2010:29). Knowledge of the characteristics of this distribution can thus be used to determine whether information is hidden in an image by comparing the histogram of a stego image against the distribution of an image with no embedded information. Since JPEG steganography replaces bits, the histogram deviates from the norm and the presence of embedded information can be statistically detected (Westfeld 2001:291).

To increase the statistical undetectability of JPEG steganography, the Outguess and F5 algorithms were developed to offer a higher level of resistance against statistical attacks.

4.3 Outguess

The Outguess algorithm was first introduced by Provos (2001:323) and is seen as an example of a steganographic algorithm that performs statistical restoration. LSB embedding is used to embed message bits into the LSBs of DCT coefficients, except the coefficients zero and one (Li et al 2011:150). Zero and one are skipped to avoid visible artefacts in the histogram (Fridrich 2010:108). After embedding, corrections are then made to the DCT coefficients that were not used for embedding to match the histogram of the coefficients of the stego image with the histogram of the cover image (Provos 2001:323).

Before embedding, the algorithm calculates the maximum length of a secret message that can be accommodated, while ensuring that there are enough unused coefficients for the correction phase (Fridrich 2010:108).

4.4 F5

The F5 algorithm was developed by Westfeld (2001:289) and also focuses on preserving the histogram of a stego image. During embedding, instead of changing the LSBs of coefficients whose LSB does not match the message bit, the F5 algorithm decrements the absolute value of the coefficient by one (Li et al 2011:150). Coefficients equal to one are not used for embedding to avoid visual distortion (Fridrich 2010:120). By not overwriting the LSBs of coefficients, changes to the cover image are no longer visible in the stego image's histogram (Westfeld 2001:300).

To increase the payload capacity and further increase the statistical undetectability of the F5 algorithm, matrix embedding is employed to decrease the number of changes made to the cover image (Westfeld 2001:297).

5. EVALUATION OF THE IMAGE STEGANOGRAPHY ALGORITHMS

Section 2 discussed four evaluation criteria for image steganography algorithms: invisibility, payload capacity, robustness against image manipulation attacks and statistical undetectability. During the discussions of the different image steganography algorithms in sections 3 and 4, the level in which the algorithms complied with the criteria was discussed. This section provides a summary of the evaluation of seven image steganography algorithms: LSB embedding, ± 1 embedding, LSB embedding in palette based images, optimal parity embedding, JPEG steganography, Outguess and F5.

The following sections, sections 5.1 to 5.4, evaluate the algorithms according to invisibility, payload capacity, robustness against image manipulation attacks and statistical undetectability respectively. Section 5.5 gives a summary of the comparison of the different image steganography algorithms.

5.1 Invisibility

The level of invisibility is high in all of the above image steganography algorithms, with the exception of LSB embedding in palette based images. In LSB embedding in palette based images invisibility is only high if the palette is preprocessed through sorting or colour

duplication. Changes to the LSB values of palette based images without preprocessing the palette could lead to visual distortion.

5.2 Payload capacity

When comparing the payload capacity of image steganography algorithms, a distinction is made between spatial domain steganography and transforms domain steganography. The embedding rate of spatial domain steganography algorithms is measured in bpp (bits per pixel), while the embedding rate of transform domain steganography is measured in bpnc (bits per non-zero coefficient). The payload capacity of algorithms from different domains can thus not directly be compared. In the spatial domain, LSB embedding, LSB embedding in palette based images and ± 1 embedding, each have an embedding rate of up to 3 bpp. These algorithms can accommodate even more embedded information with no visual distortion if noisy images are used as cover images. However, optimal parity embedding can accommodate 1 bpp since a bit from the message is embedded in the parity bit of each pixel instead of the LSB.

In the transform domain, the maximum embedding rate that can be achieved with JPEG steganography and F5 is 1 bpnc. The payload capacity of Outguess is however more difficult to determine, since embedding can only be done on a subset of non-zero coefficients while ensuring that enough unused coefficients remain to correct the histogram of the image. The embedding rate can thus be seen as a ratio between the message length and the number of non-zero coefficients of the image and will vary from one image to another.

5.3 Robustness against image manipulation attacks

When determining the robustness of algorithms against image manipulation attacks, a distinction can again be made between images in the spatial domain and those in the transform domain. Spatial domain formats and therefore spatial domain steganography algorithms are not robust against image manipulation attacks, since changes to an image results in direct changes to the bits of the image data. Transform domain steganography algorithms, on the other hand, are robust against image manipulation attacks since image data is stored in the transform domain and not directly accessible.

5.4 Statistical undetectability

The level of statistical undetectability of an image steganography algorithm is determined by the amount of noticeable difference between a normal image and a stego image. Embedding often results in changes to the histogram of a stego image that would not occur naturally and when detected by a warden, could prove the existence of hidden information. In the case of LSB embedding in palette based images, statistical undetectability is lowered through the preprocessing of the palette. The statistical undetectability of LSB embedding, LSB embedding in palette based images, and JPEG steganography are low and thus improved algorithms were developed: ± 1 embedding improved on LSB embedding, optimal parity embedding improved on LSB embedding in palette based images and both Outguess and F5 were developed as improvements on JPEG steganography. The four improved algorithms were developed with the goal of minimising the signature left on the image histogram through embedding and thus the statistical undetectability of these algorithms is high.

5.5 Summary of image steganography algorithm comparison

Table 5.1 shows a comparison of image steganography algorithms. For each algorithm, the level with which the algorithm complies with the evaluation criteria is indicated as high or low. Payload capacity gives the embedding rate.

As seen from Table 5.1 a trade-off often occurs between payload capacity and statistical undetectability. More advanced algorithms often provide more resistance against statistical attacks, but at a loss of embedding capacity. To embed a message in an image without leaving a signature can only be done successfully with smaller messages.

6. CONCLUSION

This chapter did not provide a complete list of all possible image steganography algorithms, but instead provided a list of well-known algorithms that were deemed relevant for a discussion on image steganography in secure communication.

The focus of this chapter was on different image steganography algorithms and how they are implemented. Through knowledge of the functionality of image steganography algorithms,

Table 5.1. Comparison of image steganography algorithms

	LSB embedding	± 1 embedding	LSB embedding in palette based images	Optimal parity embedding	JPEG steganography	Outguess	F5
Invisibility	High	High	High*	High	High	High	High
Payload capacity	Up to 3 bpp	Up to 3 bpp	Up to 3 bpp*	1 bpp	1 bpnc	Varies depending on image	1 bpnc
Robustness against image manipulation attacks	Low	Low	Low	Low	High	High	High
Statistical undetectability	Low	High	Low*	High	Low	High	High

* if palette was preprocessed

informed decisions can be made on the suitability of algorithms for specific secure communication applications. The evaluation of image steganography algorithms done in this chapter can also be used to assist in the decision making process.

The first half of this dissertation, including this chapter, has focussed on studying image steganography in detail and available literature was reviewed to provide an indication of the state of the technology. Implementing image steganography in secure communication scenarios is the focus of the next chapters. With reference to the research objectives stated in chapter 1, the following secure communication scenarios need to be addressed:

- Self-communication
- One-to-one communication
- One-to-many communication

The next chapter discusses self-communication.

CHAPTER 6

SELF-COMMUNICATION

1. INTRODUCTION

Secure self-communication is the storage and retrieval of digital information, such as banking and other personal details, in a way that complies with the following requirements for a secure communication system: the communication should not be suspicious, the confidentiality of the information should be ensured, the system should keep within legal bounds and the system should be easy to use. For a user to securely access secret information, the information should ideally be stored locally, since accessing it from a remote location could open the door to eavesdropping attacks. It should thus be stored on a device that can travel with the remote user.

A challenge of secure self-communication is to ensure the confidentiality of the stored information, even if an unauthorised person were to obtain the information, for example by stealing the device that the information is stored on. Encryption could normally be used to encipher the secret message so that the encrypted information can only be deciphered with a secret key. However, due to the appearance of encrypted information and legislation, encryption does not comply with the requirements of inconspicuousness and legality. Since steganography also offers confidentiality of information, image steganography is thus implemented as an alternative.

The secure communication approach to self-communication proposed in this chapter uses image steganography to store information by hiding it in images. The information can then only be retrieved if it is known that the images contain the hidden information and the method with which it was embedded.

Mobile devices, such as mobile phones and digital image viewers, are ubiquitous and can serve as storage devices for sensitive information. Digital cameras and digital image viewers can store images, while devices such as smart phones not only store the images, but also possess processing power. Hiding information in images using image steganography and then storing the images on a mobile device provides portability and confidentiality.

To illustrate an application of image steganography for secure self-communication, a password system is used as an example. Passwords are frequently used to protect access to computer systems and remote users typically have different passwords for different systems, to be stored and retrieved when necessary. Passwords, however, suffer from a number of drawbacks – notably the fact that they can be compromised by sniffing or other means (McDonald, Atkinson & Metz 1995:16). One particularly widespread attack is known as the replay attack (Syverson 1994:187). A replay attack involves passive capturing of passwords through eavesdropping and if the password has not changed since last used, it enables the eavesdropper to reuse the password for authentication.

A common mechanism to guard against replay attacks is the use of one-time passwords (Lamport 1981:770; Jeong, Chung & Choo 2008:295), where a single password is used only once for authentication. To enable the secure storage and retrieval of one-time passwords in secure self-communication, this chapter proposes using image steganography to hide one-time passwords in images that are stored on a mobile device. The proposed system offers resistance against replay attacks by using one-time passwords and offers confidentiality by hiding the existence of the one-time passwords. Portability is offered through the use of mobile devices.

The rest of the chapter starts by first reviewing current one-time password systems and their possible vulnerabilities. An overview of the proposed system is given in section 3 with more technical details discussed in section 4. An evaluation of the system to determine whether the system complies with the requirements of a secure communication system is done in section 5.

2. ONE-TIME PASSWORDS

According to McDonald et al. (1995:16) the following two conditions should be met when implementing a one-time password system:

1. Even if an eavesdropper should intercept a one-time password communicated over a public channel, access to the target system should not be possible

2. Should a one-time password be compromised by an unauthorised party, the unauthorised party should not be able to deduce the next password from the previous password.

A popular and widely used one-time password system is S/Key. The system was originally described by Haller (1994:151) and has since undergone updates to its current standard. The S/Key system has been modified by the U.S. Naval Research Laboratory and the modified system is referred to as OPIE (One-Time Passwords in Everything) (McDonald et al 1995:16).

One-time password (OTP) systems can be divided into three groups, depending on how the OTPs are generated: mathematical algorithm OTPs, time-synchronised OTPs, and challenge response OTPs. The original OTP system developed by Lamport (1981:770) made use of a mathematical algorithm and a one-way function to encode the passwords, effectively creating a password chain where the previous passwords are used in the calculation of the next password. S/Key and OPIE make use of mathematical functions (Haller 1994:151; McDonald et al 1995:16).

Time-synchronised OTPs are usually related to special electronic tokens (Choi & Thang 2010:91) where a password is generated by the token based on the current time. Challenge response OTPs use mathematical algorithms and the new password is based on a challenge, for example a random number generated by the authentication server (Pfleeger 2000:262; Tsai 2003). Electronic tokens can also be used for challenge response OTPs where a challenge is encrypted with a secret key programmed into the token (Chapman & Zwicky 1995). The authentication server encrypts the challenge with the same key and the user is successfully authenticated if the user's cipher is the same as the authentication server's cipher.

Although these three OTP generation methods have been around for some time, methods for storing and displaying the next OTP to the user are continuously evolving. Research that has been done on these methods divides OTP systems into systems where the OTP is generated by the user and systems where the OTP is generated by the authentication server and then communicated to the user.

Systems that use special electronic tokens are typical examples of systems where the token is generated by the user. Special electronic tokens, for example technology company EMC's RSA SecureID token, generate a time-synchronised OTP and display the password on a small screen (RSA SecureID, www.emc.com/security/rsa-securid.htm). Electronic tokens, however, have a number of drawbacks: with time-synchronised OTP systems the clock of the token and the clock of the authentication server can drift out of sync, usually due to low battery power on the token side (Jorns, Bessler & Pailer 2005:65). Electronic tokens typically have a short battery life (Valente, Redd & Northcutt 2009:2) that results in either the battery or the entire device to be replaced periodically. This makes the token system more costly and potentially unreliable when travelling. Additionally, users have to carry and protect yet another electronic item that may be lost or stolen.

Another, more cost-effective, way for the user to generate the OTP on demand is downloadable software that runs on a mobile phone and can generate OTPs (Aloul, Zahidi & El-Hajj 2009:641; Prakash, Infant & Shobana 2010:133). SolidPass is one such system where software tokens are installed on a mobile device to generate challenge response OTPs (SolidPass website, www.solidpass.com).

OTPs that are generated by the authentication server can either be generated once-off as a password list that is kept by the user or the passwords can be generated on demand and communicated to the user. Traditionally, OTP lists were printed out on paper that the user had to carry with him (Halevi & Krawczyk 1999:235). Clearly, if such a list was lost or stolen, the security of the system would be compromised.

More recent research has provided a solution where an OTP is generated by the authentication server and communicated to the user via an alternative channel, such as short message service (SMS) (Florêncio & Herley 2008:401). However, the encryption provided by most service providers are often not sufficient (Croft & Olivier 2005:71) and messages can be intercepted (Lo, Bishop & Eloff 2008:154), making the system vulnerable to man-in-the-middle attacks. Additionally, the mobile phone service provider becomes part of the circle of trust and as is the case when travelling abroad, more than one service provider has to be trusted to enforce a specified security policy (Grandison & Sloman 2002:145).

However, all of these methods for displaying OTPs to the user make it obvious that OTPs are being used and information on how the OTP system is implemented can be deduced by an unauthorised party. Access to this information does not comply with the requirement for inconspicuousness. In the proposed system, the OTPs are generated by the authentication server, but instead of communicating the OTPs to the user, the passwords are hidden in images on the mobile device where they can be accessed when needed. The proposed system is thus concerned not only with storing the OTPs, but also with hiding their existence. The next section provides an overview of the proposed system, the Stego-OTP system.

3. OVERVIEW OF THE STEGO-OTP SYSTEM

This section gives an overview of the Stego-OTP system and identifies aspects of the system that needs to be discussed in more detail in section 4.

The main idea behind the Stego-OTP system is to enable self-communication by using image steganography as a means of hiding one-time passwords on a mobile device, thereby enabling users to inconspicuously carry a pre-generated list of one-time passwords with them when travelling. Based on the assumption that the average person would not suspect this additional use of images, the system not only protects the confidentiality of the OTPs, but also the fact that they exist.

The design of the proposed system uncovered a number of questions that should be taken into consideration. All of these questions are examined in more detail in the next section. The first question is which system to use for OTP generation. Since the OTPs in the Stego-OTP system should be generated by the authentication server, the initial choice is to use a mathematical OTP system to generate the passwords. However, with mathematical OTPs there is a definite order in which the passwords are generated and used. For the proposed system this pre-defined order does not necessarily add to the security of the system and the use of random passwords are thus explored in section 4.1.

The choice of image steganography algorithm is another aspect to consider and is discussed in section 4.2.

Once the OTPs have been embedded in the images and stored on the mobile device, the next question is how the user should select an image to extract the OTP from. Since the images are not stored sequentially, there is no order in which they have to be used. However, complete randomness can be refined to provide extra functionality to the system and is discussed in section 4.3.

The final question is how the OTPs should be displayed – in fact communicated – to the user. This entails the extraction of the OTP from the image and depends largely on the type of mobile device used. Possible approaches to extracting OTPs are examined in section 4.4.

An overview of the Stego-OTP system is thus as follows: Before travelling, a list of n OTPs are generated and embedded in n images. These images are then stored on a mobile device. When needed, the user selects an image (Image_x), extracts the OTP from the image (OTP_x) and uses the OTP for authentication. This process is illustrated in Figure 6.1. During the storage phase the passwords are generated and embedded and during the retrieval phase the image is selected and the password extracted.

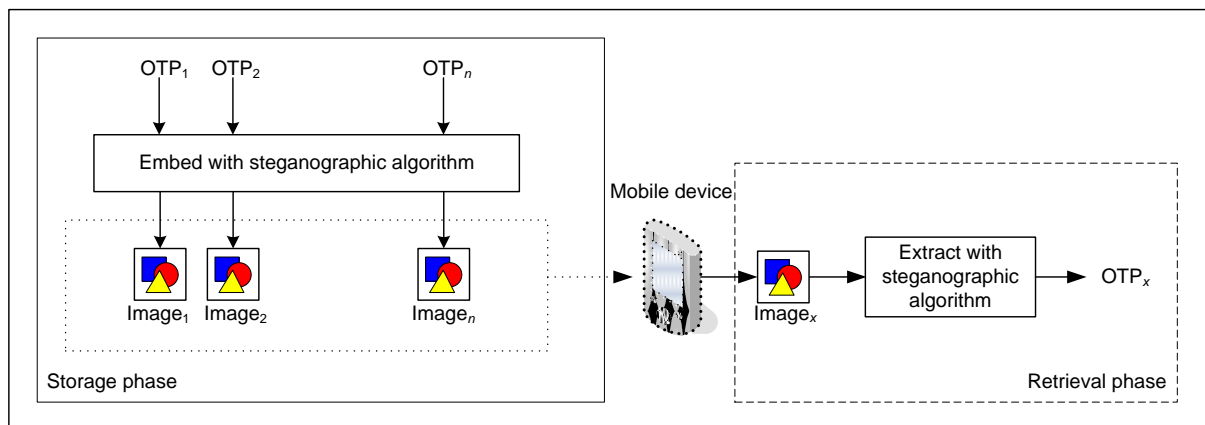


Figure 6.1. The Stego-OTP system

The next section discusses the design of the Stego-OTP system in more detail.

4. DESIGN OF THE STEGO-OTP SYSTEM

This section builds on the overview of the Stego-OTP system given in section 3 and offers a more detailed explanation of certain aspects of the system. Recommendations for the generation of the OTP list is given in section 4.1. Section 4.2 discusses the selection of an

image steganography algorithm and section 4.3 discusses the selection of an image. The extraction of the OTP is discussed in section 4.4. The storage phase is discussed first and starts with how the OTP list should be generated.

4.1 Generating the OTP list

Existing techniques for generating OTP lists without the help of an electronic token use mathematical functions to create a password chain. A mathematical OTP chain has a pre-defined order in which the passwords should be used for authentication. Generating a list of OTPs that has to be used in a specific order is not necessarily the most suitable approach to the Stego-OTP system since images are often not numbered according to a specific order. Numbering the images sequentially could create suspicion, since sequential image numbers occurring in a real-world scenario are unlikely. A sequential list of passwords additionally does not add to the security of the proposed system, since once a password has been used, it is redundant and should be deleted. A suspicious pattern in the deletion of images could give a clear indication of the next password to use in the chain.

Generating random passwords is thus recommended. In creating random passwords, assume that a users' list has n passwords and that these passwords are selected from a space of M possible passwords. The probability of guessing a correct password (or encountering it during a brute-force attack) is $1/M$. If random unused passwords can be used, the probability of this happening is initially n/M and decreases as passwords are used. This does not significantly affect system security as long as $M \gg n$.

A list of randomly generated passwords is thus created by the authentication server to be stored on the mobile device. The size of the password list should be proportional to the number of anticipated authentication requests to be made by the user. Once the passwords have been generated, they are embedded in the images using image steganography.

4.2 Selecting an image steganography algorithm

The most important characteristic of the selected image steganography algorithm is its compatibility with image file formats that are used on mobile devices. Most mobile devices, such as digital image viewers, cellular phones and PDAs, use the JPEG image file format.

Not only does the JPEG compression algorithm result in smaller image sizes, but JPEGs are also very successful in representing photographic images of real-world scenes (Moerland 2003). Most digital cameras and cellular phones with photographic cameras also store photos in the JPEG format.

A large payload capacity is not a major concern when selecting the image steganography algorithm, since an OTP can generally be stored in a few bytes of data consisting of a few characters and numbers. However, a high level of invisibility is important in order to hide a password in an image. Taking these aspects into consideration, the recommended image steganography algorithm is JPEG steganography.

When the user wishes to use an OTP for authentication, a suitable image should first be selected and then the OTP is extracted from the image. Selecting images that are suitable for use as cover images are discussed in the next section as well as selecting an image for password extraction from the user side.

4.3 Selecting an image

In the storage phase, the images to use as cover images should first be selected based on their suitability. For example, images should be in the correct image format for the selected image steganography algorithm and have the necessary embedding capacity for the OTPs. To reduce suspicion, images should be of someone or something that has a reason to be stored on the mobile device, for example, the user's family. For reasons of inconspicuousness it would thus be best if cover images are selected from existing images stored on the mobile device. If not possible, or if the mobile device does not contain enough images, cover images can be selected from a pool of suitable images.

Once a set of suitable images have been identified, a simple approach is to embed n OTPs in n images and store only these n images on the mobile device. In the retrieval phase the user can then randomly select an image from the mobile device and extract an OTP. However, the implication of this approach is that all of the images on a mobile device contain embedded information. When considering the possibility that the device might be lost or stolen and the further possibility that the person who took the device might know what the images are used

for, hiding information in all of the images would result in a 100% probability of the attacker accessing an OTP.

Only a selection of images on the mobile device should thus contain embedded information and the user should be able to differentiate between images that do contain passwords and the ones that do not. The most logical approach to differentiate between stego images and images with no embedded information, is to make use of the visual characteristics of images to divide them into categories according to theme. Themes can be based on subject matter (such as landscapes vs. other images, portraits vs. other images, portraits of family members vs. other images), colour, orientation or any other conceivable ‘theme’. In the retrieval phase, when selecting an image the user thus knows to only use images from one selected theme. This technique is referred to as themed recall.

Without knowledge of the correct theme an attacker may attempt to extract OTPs from images that do not contain any embedded information. An authentication attempt with an incorrect password would fail. However, failed authentication does not prevent an attacker from continuing to extract information from different images until a valid OTP is found. This brute force attempt can be prevented by adding false passwords to the system. When the authentication server generates a list of OTPs, a list of false passwords are also generated and stored in themes that the user will know does not contain embedded information. When an attacker tries to gain access to the system through a false password, the system is alerted that the device has been compromised and further attempts at authentication will be refused.

A false password is not the same as a password that is simply incorrect. Due to the possibility of human error, for example, typing errors, not every attempt at authentication with an incorrect password can be identified as a security breach. However, when someone who knows how to extract an OTP, but extracted it from an incorrect image, attempts to authenticate with a false password, the device should be considered as compromised.

Once cover images have been selected based on suitability and theme, the OTPs are embedded in the images with the image steganography algorithm and stored on the mobile device. In the retrieval phase, when an OTP is required, a suitable image is selected based on themed recall and the OTP is extracted from the image and used for authentication. Methods for extracting the OTPs are discussed in the next section.

4.4 Extracting the image

Mobile devices that do have processing power often support Java 2 Micro Edition (J2ME) or a similar platform, enabling the devices to execute customised applications. The extraction of the OTPs can then be done on the device itself. This is the most secure form of self-communication since the information is only communicated between user and storage device.

Shirali-Shahreza (2006) and Papapanagiotou et al (2005:589) have both successfully implemented image steganography on cellular phones. Although Shirali-Shahreza (2006) only implemented LSB steganography to embed information in black and white images used in SMS's, Papapanagiotou et al (2005:589) indicated that the overall performance of JPEG steganography on a cellular phone is satisfactory.

However, a minority of mobile devices have limited or no processing power and it will not be possible to extract the OTP from the device while the image is stored on the device. In these cases, a user has to make use of an auxiliary computing device in order to facilitate communication between the user and the storage device. The image has to be communicated to an external party, via a potentially insecure computer device, from where the password can be retrieved.

Two options for extracting the OTP password from a device with limited or no computational power are (1) to input the image to a secure web application that will extract and return the OTP or (2) to download extraction software to the auxiliary computer device.

The main risk of the first option, as illustrated in Figure 6.2, is that an eavesdropper may intercept the stego image while it is being communicated, extract the password, and use it in a replay attack, since it has not yet been used for authentication by the user. This nullifies the advantages of using OTPs, thus proving the first option to be an unacceptable solution.

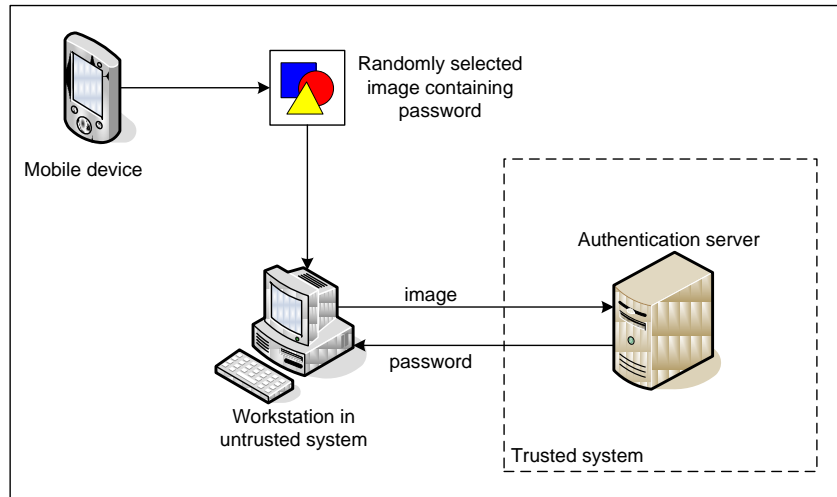


Figure 6.2. A scenario where the image is transmitted through an untrusted network to a secure server. The application on the server extracts the password and sends it back to the workstation.

The second option solves the problem of eavesdropping by not communicating the password back and forth, but rather acquiring the software in order to extract the OTP. Since the mobile device cannot execute the software itself, a potentially insecure computer device has to be used. This option is illustrated in Figure 6.3.

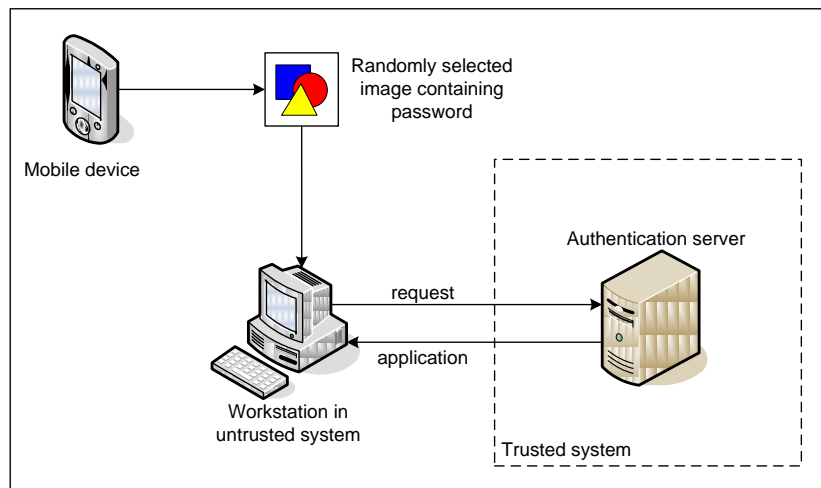


Figure 6.3. A scenario where the client workstation issues a request to the secure server. The server transmits the extraction application to the client. The image and password is never communicated over the network.

The main vulnerability with this option is that an eavesdropper might intercept the application as it is being transmitted, thus acquiring the means for extracting all subsequent passwords on the mobile device. However, this only exposes the system when the same

eavesdropper has also managed to steal the mobile device that the passwords are stored on, making this option slightly more acceptable than the first.

Considering these options it is thus apparent that when the mobile device has limited or no processing power, the best option still results in opening the self-communication to the risk of eavesdropping attacks. Using a mobile device with processing power is thus recommended.

5. EVALUATING THE STEGO-OTP SYSTEM

This section examines the proposed Stego-OTP system to determine whether the system complies with the requirements of a secure communication system as stipulated in chapter 1, namely: inconspicuousness, confidentiality, legislation, and ease of use.

Communicating the OTP to the user is done as easily as it would have been using other one-time password systems. Instead of using tokens or alternative channels to display the generated OTP, the user need only use software stored on the mobile device (or an auxiliary device) to extract the password from the image. The main advantage of the proposed system over other one-time password systems, is essentially the combination of the separate advantages offered by each of the underlying technologies.

The inconspicuous transport of OTPs is the main advantage offered by the inclusion of image steganography in the system. Carrying a list of generated OTPs with the user has its obvious disadvantages since the list can be lost or stolen, however carrying a physical token with which to generate passwords on demand also has disadvantages, mainly that the presence of the tokens reveal the presence of OTPs. Systems where the OTP is communicated to the user when needed has the main disadvantage of an additional communication channel to be trusted and kept secure. Image steganography hides the existence of the OTPs, thus providing inconspicuousness.

The confidentiality of the OTPs is also ensured by the use of image steganography and since encryption is not used, the system is also within the bounds of international laws.

The advantage of using mobile devices to store the OTPs on, lies in the fact that mobile devices are ubiquitous in the sense that almost everyone has one and carries it with them

everywhere. This means that the presence of a mobile device does not inconvenience the user, is not suspicious, and a potential attacker will not necessarily foresee the real use of the images stored on the device.

Finally, it is important to note that the proposed system is used as a system for self-communication and is not intended to protect the user against keyloggers or the passive capturing of passwords. However, the fact that the passwords are one-time passwords offers protection against these attacks.

6. CONCLUSION

This chapter proposed using image steganography to hide one-time passwords in images as a means of secure self-communication. The Stego-OTP system complied with all of the requirements of a secure communication system from chapter 1, thus it can be deduced that image steganography can successfully be used as an alternative to cryptography in this scenario. The next chapter discusses using image steganography for one-to-one communication.

CHAPTER 7

ONE-TO-ONE COMMUNICATION

1. INTRODUCTION

Secure one-to-one communication refers to the communication between one sender and one receiver. For the purposes of this chapter, it is assumed that the remote user acts as the receiver. When the receiver has access to a computer device that can be trusted, the receiver can easily store and use the software necessary for the communication to take place. However, when the receiver does not have access to a trusted computer device, a publicly available untrusted computer will have to be used with the risk of Trojan horses and other keyboard sniffing software (Oprea et al. 2004:438). The communication channel is also not without risks, since the communication can be intercepted during transit (Borisov, Goldberg & Brewer 2004:77).

One-to-one communication thus has two potential vulnerabilities: at the sender and receiver endpoints and during transit. Firewalls and network security can be implemented to secure the endpoints (Borisov, Goldberg & Brewer 2004:77), while security protocols using encryption can be used to secure the communication in transit. However, the legality of encryption in certain countries can limit its use.

Pretty good privacy (PGP) developed by Zimmermann in 1991 (Garfinkel 2003:1421) is an example of a security protocol that uses encryption to secure electronic mail (e-mail). Security protocols, such as PGP, are however not without vulnerabilities of their own. A well-known attack on e-mail security protocols is an adaptive chosen ciphertext attack (Katz & Schneier 2000:18), where an attacker submits his own encrypted message to a decryptor in order to get enough information to decrypt intercepted messages. At first thought to be only a theoretical attack, Jallad et al (2002:90) implemented the chosen ciphertext attack successfully on PGP.

Further vulnerabilities of PGP lie in the cryptographic key size. Although the key sizes used by PGP offer an acceptable level of security (Lentra & Verheul 2001:256), the passwords that are used to protect the private PGP keys do not. When encryption is implemented to protect communication, it also does not hide the fact that communication is taking place which does

not comply with the requirements for a secure communication system as discussed in chapter 1.

Image steganography, on the other hand, not only protects the contents of the communication, but also the existence of it. This chapter proposes a secure one-to-one communication system that uses image steganography to hide information in images before the images are communicated by the sender to the receiver. As an example of where image steganography can be implemented to communicate sensitive information, the scenario of decryptor distribution is used.

The rest of this chapter is outlined as follows: Section 2 explains what decryptor distribution is. The proposed decryptor distribution system is discussed in section 3, followed by an implementation of the system in a prototype in section 4. Experimental results obtained with the prototype are presented in section 5 along with a discussion on suitable bit depths for cover images.

2. DECRYPTOR DISTRIBUTION

As discussed in chapter 1, encryption is a popular technology when communication must be done in private. Consider a scenario where a remote user is travelling from one country to the next. Some of these countries may allow encryption while others may not. When the user has access to a trusted computer device, the necessary decryption software – called the decryptor – can be stored on the device. However, since even the possession of encryption software is sometimes illegal (Dunbar 2002:2) the user will need to delete the decryptor when entering such a country and will only be allowed to install the software again after leaving that country.

However, even when encryption is legal the distribution of the decryption software is complicated since the remote user may need to acquire and store several different decryptors for communication protocols with different people. To ensure that the receiver has stored all the necessary decryptors before travelling, implies that the receiver knows beforehand which decryption software will be needed, which is not always the case. In certain scenarios the receiver may only find out that a specific decryptor is required while travelling. A secure system for distributing the decryption software is thus needed.

A system for decryptor distribution should comply with the same requirements as any other secure communication system, namely inconspicuousness, confidentiality, legality and ease of use. To ensure the legality of the communication system, the system proposed in this chapter can only be implemented in countries where encryption is legal. If encryption is not legal, image steganography can be used instead of encryption for secure one-to-one communication. In the proposed system, confidentiality of the secret information can be achieved through the inclusion of encryption. To achieve inconspicuousness, the decryptor as well as the message is hidden in an image, therefore hiding the fact that communication is taking place.

The proposed system thus needs to focus on ease of use. Ease of use can be extended by adding an extra requirement in addition to the above mentioned four requirements: For the purposes of one-to-one communication, a secure communication system should also minimise the amount of additional information about the implementation of the system needed by the remote user.

If the remote user needs to obtain additional information in order to use the decryptor, the communication of the additional information could make the decryptor distribution system vulnerable to interception attacks. The system proposed in this chapter, uses image steganography to embed not only the encrypted message, but also the decryptor software for decrypting the message, in an image. The decryptor can thus be distributed on demand.

3. THE DECRYPTOR DISTRIBUTION SYSTEM

The decryptor distribution system suggests treating the stego image not as a single object, but as a container for two objects: the software needed for decrypting a message and the message itself. More than one object is thus embedded in the stego image. Different types of information objects can be embedded in an image and still only a single image is communicated between sender and receiver. For decryptor distribution, the encrypted message and the decryptor software are the two objects that are embedded in the image. For ease of use and to minimise the amount of additional information, the message is encrypted using a symmetric encryption algorithm, as discussed in chapter 2, since symmetric encryption algorithms require only one key. The encrypted message along with the decryptor software to decrypt the message is then embedded in the cover image.

Normally the decryptor would only be responsible for decrypting the encrypted message. However, to comply with the requirement of minimising additional information, it is proposed here that the decryptor be responsible for extracting the message as well. Ideally, different image steganography algorithms should be used for embedding the message and the decryptor so that an attacker would not automatically be able to extract the one because the other could be extracted. However, different algorithms would imply that the receiver has to have knowledge of both extraction methods which again increases the amount of information the user will need to use the system. Using the decryptor to extract the message means that the receiver need only know how to extract the decryptor.

The system is divided into two phases: the embedding phase and the extracting phase. These phases are illustrated in Figure 7.1.

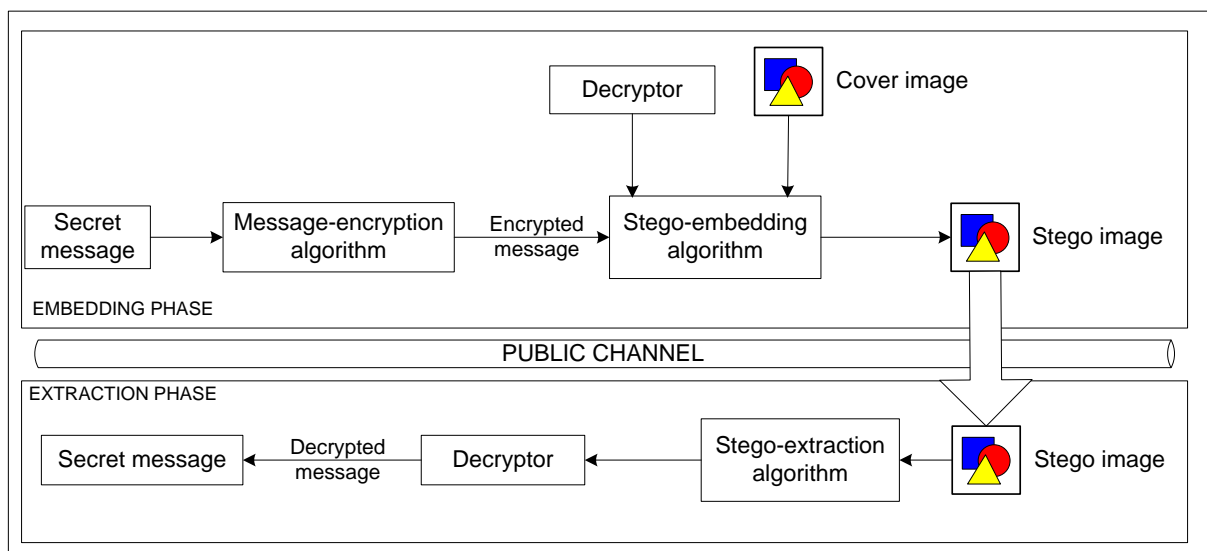


Figure 7.1. Process diagram of the decryptor distribution system

Section 3.1 discusses the embedding phase of the decryptor distribution system and section 3.2 discusses the extraction phase.

3.1 The embedding phase

The embedding phase is responsible for encrypting the secret message and embedding the information in the cover image. The next section discusses the algorithms used in the embedding phase to encrypt and embed the message. Section 3.1.2 discusses the format of

the decryptor, section 3.1.3 explains the use of a decryptor header and section 3.1.4 discusses the LSB embedding algorithm used in the decryptor distribution system.

3.1.1 Algorithms used in the embedding phase

The embedding phase consists of a message-encryption algorithm to encrypt the message and a stego-embedding algorithm to embed the message in the cover image.

The message-encryption (ME) algorithm is used to symmetrically encrypt the secret message. Two factors that play an important role in the choice of encryption algorithm, is the size and complexity of the resulting program. Not only will an elaborate and complex algorithm require a larger cover image, but the processing power required to execute such a complex program could limit the functionality of the system by eliminating devices with limited processing power, such as mobile devices. The encryption algorithm is however, not the focus of this chapter.

The stego-embedding (SE) algorithm forms part of the image steganography algorithm and deals with embedding the information in the cover image. If different image steganography algorithms are used for embedding the message and the decryptor, one part of the SE algorithm will be used for embedding the message and the other part for embedding the decryptor. The same algorithm can also be used, but with different parameters, for example the decryptor can be embedded in adjacent pixels of the cover image, while the embedding of the message can be done randomly.

Both parts of the SE algorithm require invisibility of the embedded information. For the decryptor part of the SE algorithm, a high payload capacity is an additional requirement, since two information objects will be embedded in the cover image. The extraction part of the image steganography algorithm that embeds the message should not be a complex algorithm, since a complex extraction algorithm could result in a larger decryptor file size.

LSB embedding in BMP images has a high payload capacity as discussed in chapter 5. LSB embedding is also a simple algorithm, both for embedding and extraction, making it also suitable for embedding the encrypted message. LSB embedding is thus recommended for the decryptor distribution system.

3.1.2 *The decryptor format*

Although the decryptor is not explicitly used in the embedding phase, the decryption software has to be embedded in the cover image along with the secret message. The decryptor can either be source code or an executable program.

The main advantage of source code is that the receiver is able to examine the implementation of both the extraction of the message as well as the decrypting of the message. This is advantageous if knowledge of the algorithms should also be transferred for further communication between the sender and receiver following the same communication protocol. Another advantage of the receiver being able to examine the source code before executing it, is the risk of a malicious program sent by an untrusted sender. Through examination of the code, the user can determine whether or not it is safe to execute the program.

On the other hand, should the relationship between the sender and receiver call for the confidentiality of the decryptor from the receiver himself, an executable program would be best since attempting to decompile an executable program is a complicated task (Cifuentes & Gough 1995:811). It is, however, not impossible to decompile an executable program back into source code, but the decompiling process takes time and effort and could at least deter a party from reverse engineering the software. With an executable program, the receiver can only execute the program without knowing how the information is extracted or decrypted.

The choice between source code and an executable program thus depends on the nature of the relationship between the sender and receiver. In the case of mutual trust between sender and receiver, either source code or an executable program can be used. However, in the absence of trust it is in the best interest of the sender to use an executable program if the decryptor algorithm should be kept from the receiver. Similarly, it is in the best interest of the receiver to communicate source code since an executable program could be harmful if sent from a malicious sender. The requirements and level of trust between sender and receiver thus dictates the decision.

3.1.3 *The decryptor header*

In an ideal world the receiver will have knowledge of the type of information to expect before communication takes place. Information such as the size of the decryptor and the file type is necessary to the receiver side for successful extraction. Communicating this information to the receiver could however, increase the risk of man-in-the-middle attacks. Additional information needed by the receiver should thus be kept to a minimum. However, it is not possible that the receiver need no knowledge of the system beforehand. Even while knowing the image steganography algorithm, the receiver will only be able to extract a random string of data without knowledge of the order of the embedded information and a few additional details.

The file size of the decryptor is one such detail that the receiver will need to know in order to know how many bits to extract and allocate to the decryptor. A possible approach for the receiver to acquire the file size is to hardcode the file size as a variable into the program used for extracting the decryptor. However, hardcoding limits the re-usability of the steganography algorithm if the extraction program is be used for future communications where the same decryptor is not used. A decryptor header is thus specified to act as a standard for containing additional information.

The following decryptor header is used in the proposed system:

```
<image header><filename.extension>${3 bytes decryptor file  
size}<decryptor source code/program><encrypted message>
```

The order of embedded information is as follows:

1. **Image header** – A number of bytes, for example the first 54 bytes of a BMP image, contains image header data and cannot be modified by embedding information in it. Attempts to modify the image header may result in an invalid image that cannot be displayed with image viewer applications.
2. **Filename** – The receiver needs to compile or execute the decryptor and since knowledge of the type of file to expect is not available, the receiver's ability to

successfully store the decryptor will depend entirely on the inclusion of the filename and format.

3. **Dollar sign (\$)** - The filename is followed by a dollar sign to indicate the end of the filename.
4. **File size** – Likewise to the filename, the receiver does not know how much information to extract from the image. Extracting more bits than is needed for the decryptor would add 'garbage' information to the decryptor file and would prevent the file from compiling and/or executing correctly. However, extracting too few bits would lead to an incomplete decryptor program and would also not execute successfully. Three bytes are allocated for storing the file size of the decryptor program which allows for a maximum decryptor file size of 2MB. Considering that a large cover image in excess of 16MB would be required to hide a 2MB file when using an embedding rate of 3 bpp with LSB embedding, 3 bytes (24 bits) are considered sufficient. If fewer bits are used to store the file size than the allocated 3 bytes, a padding of zeros is added to preserve the format of the header.
5. **Decryptor source code/program** – The decryptor is embedded next.
6. **Encrypted message** – Finally, the encrypted message is embedded, preferably using either a different image steganography algorithm than the decryptor or a variation of the same algorithm. If the exact same algorithm is used without variation for the message and the decryptor and an attacker has knowledge of the algorithm, it would be easier for an attacker to extract both items at once. It does not decrease the ease of use of the system to use different algorithms, since the decryptor extracts the message thus the receiver need not have knowledge of the extraction algorithm for the message.

The decryptor header can either be embedded using the same image steganography algorithm than the decryptor or a different algorithm, as long as the receiver knows how to extract it.

The order of the embedded information, including the decryptor header, is shown in Figure 7.2. In this illustration, LSB embedding is used to embed the decryptor and a variation of LSB embedding is used to embed the encrypted message. Instead of consecutive LSBs, this variation of LSB embedding uses random bits following a pseudo-random path so that the message and decryptor is not embedded using the exact same method.

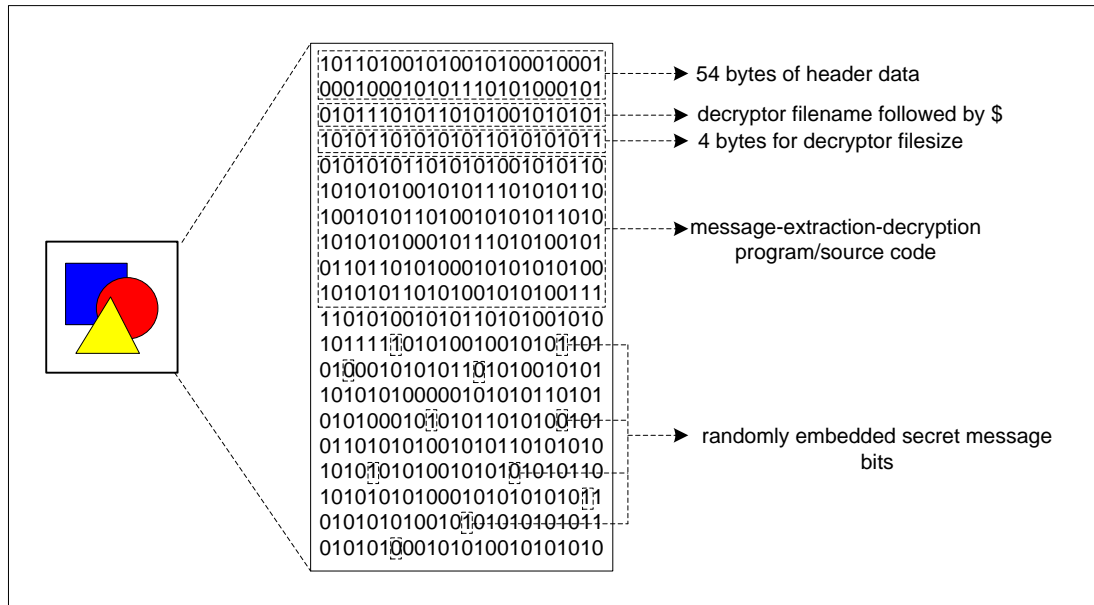


Figure 7.2. Representation of information embedded in cover image

3.1.4 LSB embedding algorithm

If LSB embedding is used as image steganography algorithm, the following algorithm for embedding the decryptor header and decryptor program in the cover image is used:

LSB-embedding algorithm:

1. Let I be the cover image with I' a representation of all the pixels of the image converted into binary. Each pixel in I is denoted as I'_i with i the pixel number, starting at 0. Each pixel consists of three colour components denoted as $I'_{i,RED}$, $I'_{i,GREEN}$ and $I'_{i,BLUE}$.
2. Let P be the decryptor and P' the binary version of the decryptor
3. Let N be the filename of the decryptor and N' the binary version of the filename
4. F is calculated as the file size of the program P' in bits and F' is the binary version of F . If the file size is less than the value stored in 3 allocated bytes, padding of zeros are added
5. To skip the BMP header and start with the 53rd pixel, set the value of i to 53
6. Starting with the first bit of N' and continuing until the end of N' , replace the LSBs of I'_i as follows:
 - Replace the LSB of $I'_{i,RED}$ with the bit from N'
 - Replace the LSB of $I'_{i,GREEN}$ with the next bit from N'

Replace the LSB of $I'_{i,BLUE}$ with the next bit from N'

Increment i

7. Convert the \$ sign into binary and let D' be the binary version of the \$ sign
8. **For** the next 8 bytes of I'
 - Replace the LSB of the next byte in I' with the first bit from D' and continue until the end of D'
9. **For** the next 24 bytes of I'
 - Replace the LSB of the next byte in I' with the first bit from F' and continue until the end of F'
10. **While** not the *end-of-file* of P'
 - Replace the LSB of the next byte in I' with a bit from P'

After the decryptor is embedded, the encrypted message can be embedded using a different image steganography algorithm. Embedding of the message is further discussed in section 4.1.

3.2 The extraction phase

At the receiver's side, the extraction phase extracts the decryptor and executes the decryptor to extract and decrypt the message. The first section discusses the stego-extraction algorithm used in the extraction phase and section 3.2.2 discusses the LSB extraction algorithm.

3.2.1 *Stego-extraction algorithm used in the extraction phase*

The decryptor is extracted using the stego-extraction (SX) algorithm. The functionality of the SX algorithm depends entirely on the image steganography algorithm used for embedding the decryptor. It is the only algorithm that the receiver should have knowledge of. However, the question is how the receiver can acquire the SX algorithm.

One option is for the sender to communicate the algorithm to the receiver via an alternative secure channel, for example e-mail. However, this approach shifts the focus from the security of the secret message to the security of the algorithm, since once an attacker can extract the decryptor, the message can also be extracted. Communication of the extraction

software between the sender and receiver could also cause suspicion and does not comply with the secure communication requirement of inconspicuousness.

If the receiver has access to a trusted computer device or storage device, the receiver can store the algorithm on the device prior to travelling. Access to the algorithm can then be obtained when needed. Mobile devices or flash drives could also be used to store the algorithm.

Finally, the receiver could write the code that implements the SX algorithm when the algorithm is needed. If a simple embedding algorithm such as LSB embedding is used and the receiver has some programming skills, the SX algorithm can be implemented by the receiver with relative ease. Since the information necessary for extracting and decrypting the message is stored in the stego image itself, the receiver need only know the format of the decryptor header and how to extract the decryptor.

3.2.2 *The LSB-extraction algorithm*

Using the same format and specifications as the LSB-embedding algorithm, the following algorithm is used to extract the decryptor from the stego image:

LSB-extraction algorithm:

1. Let I refer to the stego image
2. Set the value of i to 53
3. While the extracted value is not the \$ character
 Read the LSBs of I , 8 bits at a time
 Convert the bits into ASCII and store it in N'
4. **For** the next 32 bytes
 Read the LSBs of the next byte of I and store it in F'
5. Convert the binary value F' into an integer number F
6. **while** $F \geq 0$ do
 Read in the LSBs of I , 8 bits at a time
 Convert the bits into ASCII and store in P_x
 After each bit decrement F

7. Save P in a file called N

After extracting the decryptor, the decryptor program is compiled and executed or simply executed if not source code. The decryptor then takes the stego image as input, locates and extracts the message and, finally, decrypts it.

4. DECRYPTOR DISTRIBUTION PROTOTYPE

A prototype was developed in Java to test the proposed decryptor distribution system. The UML class diagram for the decryptor distribution prototype is shown in Figure 7.3. The next section discusses prototype considerations regarding embedding the decryptor. Section 4.2 discusses embedding of the encrypted message and section 4.3 discusses the extraction of the decryptor.

4.1 Embedding the decryptor

In the decryptor distribution prototype, the decryptor was embedded as platform independent Java bytecode (a *.class* file), mostly due to the smaller file size than source code. This, however, meant that the receiver would still need a Java compiler to execute the class. The decryptor (*Decrypt.class*) and the image file are first converted to integer arrays to represent the binary string and the image pixels. Making use of the LSB-embedding algorithm and an embedding rate of 3 bpp, each bit to embed is sent to the `changeLeastSigBit` method given in Figure 7.4. Consecutive image bytes are examined one at a time to determine whether the byte represents an even or odd number. When the number is even but the embedded bit should be a 1, the number is simply incremented and vice versa for when the number is odd.

The code segment used for LSB insertion is as follows:

```
private int changeLeastSigBit(int toAlter, char alterer) {  
    int altered = toAlter;  
  
    if ((toAlter%2) == 0) {  
        if (alterer == '1') {  
            altered++;  
        }  
    }  
}
```



```

else if ((toAlter%2) != 0) {
    if (alterer == '0') {
        altered--;
    }
}

return altered;
}

```

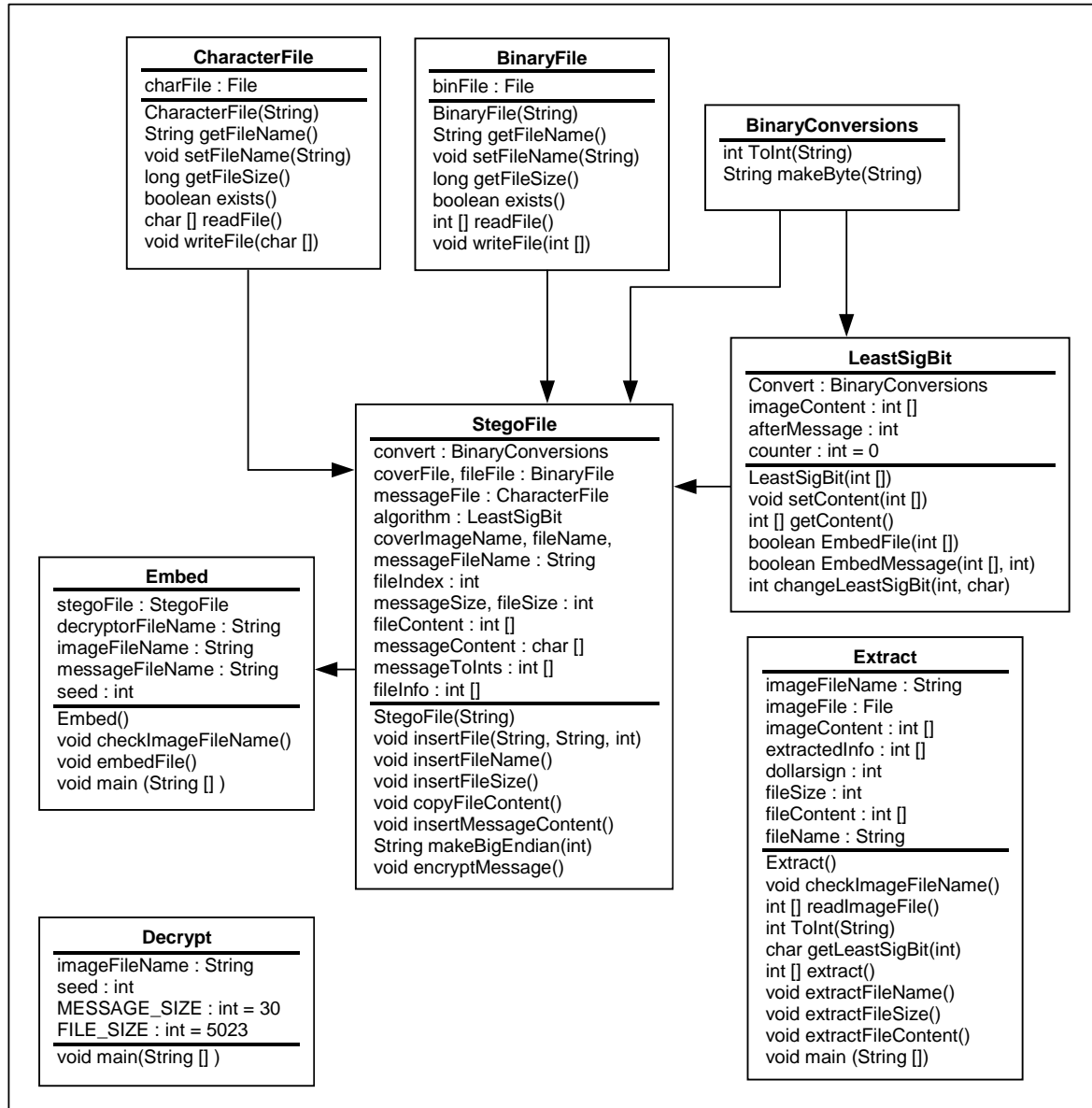


Figure 7.3. UML Class diagram of decryptor distribution prototype

4.2 Embedding the encrypted message

In the decryptor distribution prototype the simple substitution Caesar cipher (Pfleeger 2000:25) was used to encrypt the message, simply to illustrate the functioning of the

prototype. The Caesar cipher is easy to implement, but not considered to be a secure encryption algorithm (Pfleeger 2000:26). It is thus not recommended that the Caesar cipher be seen as a suitable encryption algorithm for secure communication, but merely functions as an example for the purposes of the prototype in this chapter. As discussed in section 3.1.1 a suitable encryption algorithm should be selected.

After encryption, the message is also converted to an integer array and the `changeLeastSigBit` method is again used to embed the message in the LSBs of the image. However, instead of using consecutive bytes, random bytes are selected using a pseudo-random number generator. The seed to initiate the pseudo-random numbers are shared between the sender and the receiver and without this number, the correct sequence of bytes will not be extracted. The pseudo-random number seed thus acts as secret key to the system.

4.3 Extracting the decryptor

Extract.class is an independent program not connected to any of the other classes, since it is the program that the receiver uses to extract the decryptor. After extracting the filename (*Decrypt.class*) and file size (2,672 bytes), the bits of the decryptor is extracted and stored in *Decrypt.class*. *Decrypt.class* can now be executed and stores the decrypted message in a separate file.

Figure 7.5 contains a process diagram similar to Figure 7.1, but with detailed information regarding the prototype.

5. EXPERIMENTAL RESULTS TO DETERMINE INVISIBILITY OF EMBEDDED INFORMATION

To ensure that one-to-one communication using image steganography is secure, the invisibility of the embedded information is an important requirement of the decryptor distribution system. In this section the prototype is used to determine the invisibility of the embedded information.

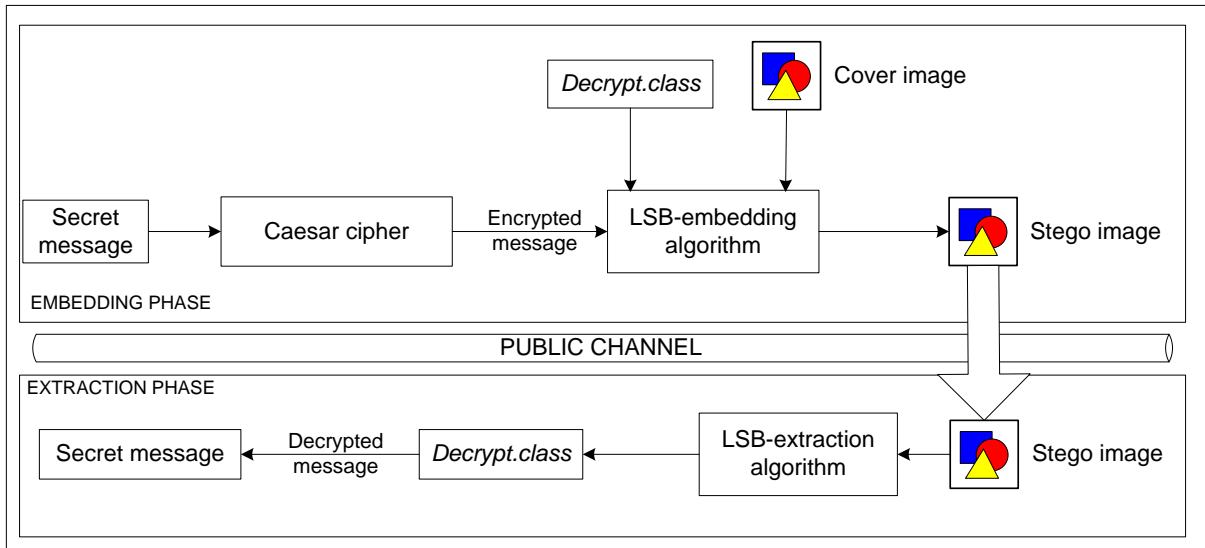


Figure 7.4. Process diagram for decryptor distribution prototype with detailed information

For experimentation with the decryptor distribution system, the prototype decryptor, *Decrypt.class*, along with an example message is embedded in an image. The size of the decryptor, *Decrypt.class*, is 2,672 bytes and the message, *msg.txt*, consists of 30 characters and is stored in 30 bytes. As example, a 24-bit RGB BMP image with image size 183KB was used. An embedding rate of 3 bpp was used for the decryptor while the message was embedded in distinct random bytes in the remainder of the image. The cover image along with the resulting stego image is shown in Figure 7.6. As can be seen, after the embedding phase, the stego image is not visually different from the cover image, thus the embedded information remains invisible.

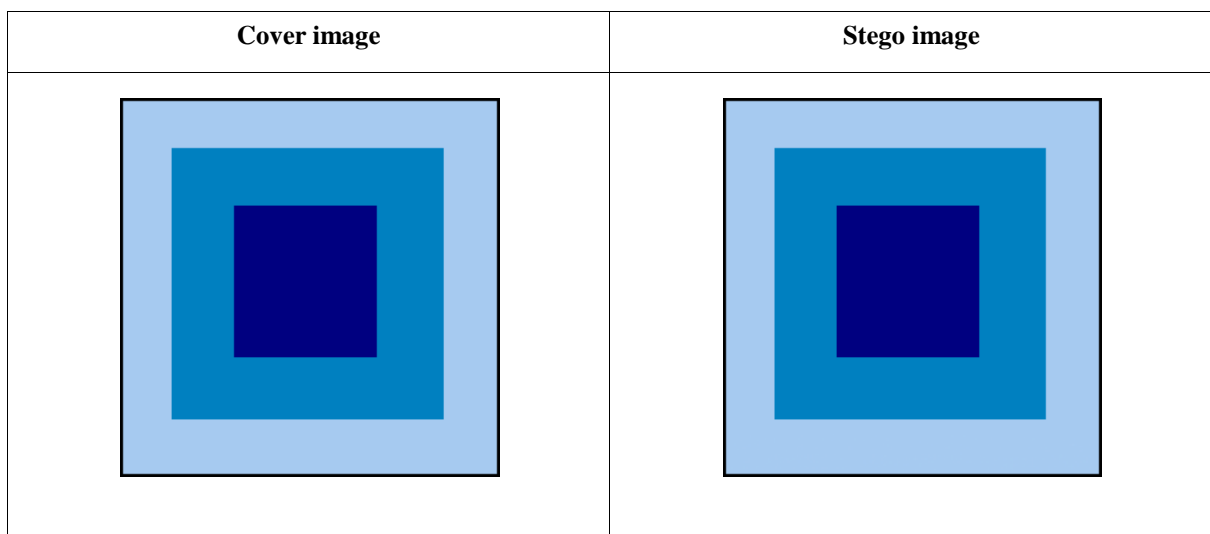


Figure 7.5. Comparison of 24-bit colour cover image, with stego image containing 2,702 bytes of embedded information

To further illustrate the invisibility of the decryptor distribution system, the same decryptor and message was hidden in a different cover image in the next experiment. In this experiment the 24-bit colour image was replaced with an 8-bit greyscale image of 63KB as cover image. The same embedding process was followed as for the previous experiment. Figure 7.7 contains a comparison between the cover image and stego image and as with the previous experiment the embedded information does not cause visual distortion.

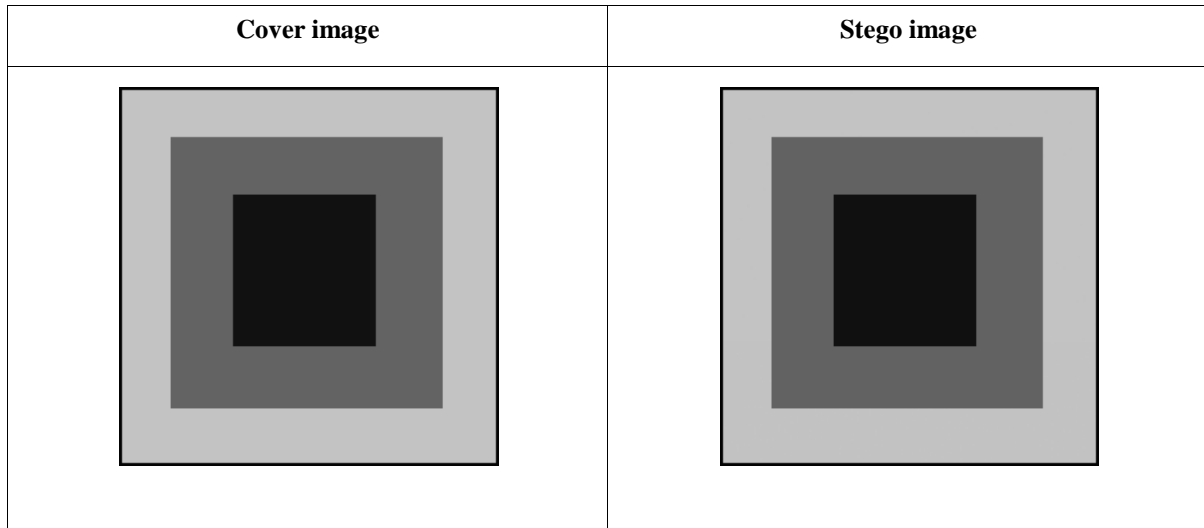


Figure 7.6. Comparison of 8-bit greyscale cover image, with stego image containing 2,702 bytes of embedded information

In another example, the same message and decryptor was again used to embed the information, but this time in an image with an even further decreased bit depth. Figure 7.8 contains the results of embedding the decryptor and message in a 32KB greyscale image with bit depth 4, in other words, an image in which four bits are used to store the colour of each pixel. Since a 4-bit greyscale image can only be represented with 16 different shades of grey, changes to the LSBs of the pixels result in larger changes to the image that can be identified visually. In the stego image displayed in Figure 7.8 the decryptor with an embedding rate of 3 bpp followed by the random distribution of the encrypted message is clearly visible. To retain the invisibility of information, the selection of suitable cover images should thus include selecting a suitable bit depth.

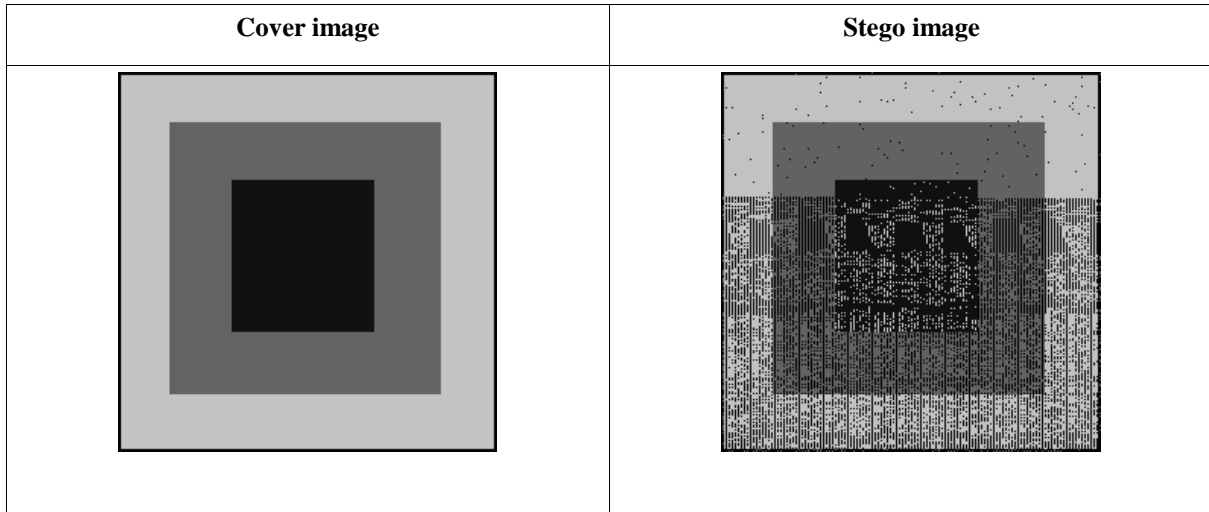


Figure 7.7. Comparison of 4-bit greyscale cover image, with stego image containing 2,702 bytes of embedded information

6. CONCLUSION

To determine whether the decryptor distribution system succeeded in qualifying as a secure one-to-one communication system, the system should be inconspicuous, confidential, easy to use and legal as specified in chapter 1. As discussed in section 1 of this chapter, the decryptor distribution system complies with legality if the system is only used in countries where encryption is allowed. The system is easy to use since it allows for the receiver to have minimal knowledge of the functionality of the system, while still being able to successfully extract and decrypt the message. The inconspicuousness and confidentiality of the system depend on the level of invisibility of the embedded information. As shown in the experimental results in section 5, neither the embedded decryptor nor the embedded message resulted in visual distortion to the image, provided that images with a large enough bit depth were used. Based on these evaluations, the decryptor distribution system does thus comply with the requirements for a secure communication system.

The next chapter discusses the development of a secure communication system for one-to-many communication using image steganography.

CHAPTER 8

ONE-TO-MANY COMMUNICATION

1. INTRODUCTION

In one-to-many communication, one sender communicates information to multiple receivers, again assuming that the receivers are the remote users. When a receiver has access to a trusted computer device, the receiver can store the software needed for the communication easily. However, since the communicated information now has multiple destinations, the risk of eavesdropping increases during transit (Dondeti, Mukherjee & Samal 2000:1681).

Secure one-to-many communication may be confused with secure group communication or secure multicasting which are communication protocols to enable communicating to multiple parties. The objectives of these communication protocols, however, differ from the objectives of the secure one-to-many communication presented in this chapter as will be seen when these technologies are discussed in section 2.

As an example of secure one-to-many communication, suppose a news media organisation wants to communicate sensitive information to its journalists in the field. The confidentiality of the information and the inconspicuousness of the communication are important, but it is also important that the identities of the participants be concealed. A journalist working on a sensitive story could be in danger should his occupation be revealed. The availability of the information is another important aspect, since multiple journalists working together, but in different locations, should all have access to the information.

Chapter 1 listed the four requirements of a secure communication system as inconspicuousness, confidentiality, legality and ease of use. A secure one-to-many communication system should comply with the first four requirements as well as the following additional requirements:

5. The system should be able to distribute the secret information to multiple parties at the same time in such a way that the information is available to all authorised parties.
6. The identities of the senders and receivers should not be easily inferred by unauthorised parties through the communication process.

This chapter proposes using image steganography to enable secure one-to-many communication by hiding information in images. To hide the existence of the communication, this chapter proposes that, instead of using point-to-point communication, the Internet is used as communication channel to add a layer of anonymity to the system. The inclusion of a public channel, such as the Internet, also ensures the availability of the information even to the possible widespread locations of the multiple receivers.

However, the vulnerabilities of a public channel results in additional mechanisms to be added to the system to ensure that the information cannot be easily accessed by unauthorised parties. This chapter thus proposes to divide the information into pieces and distribute the pieces to different locations on the Internet in such a way that the secret information can again be reassembled at a later stage. Image steganography is used to hide the pieces in images which are posted on public domain websites. Shamir's secret sharing scheme (Shamir 1979) is a simple technique used to divide a secret into a number of pieces and is discussed in the chapter.

The remainder of the chapter investigates the differences between existing group communication technologies and the proposed system in section 2. A brief overview of the proposed system is given in section 3 and Shamir's secret sharing scheme is explained in more detail in section 4. Section 5 describes the proposed system, followed by an analysis of the system's compliance to the requirements of a secure one-to-many communication system in section 6.

2. EXISTING GROUP COMMUNICATION TECHNOLOGIES

Secure group communication is a means for authorised members of a group to communicate with one another without outsiders listening in on the conversation (Aparna & Amberker 2007:359). Secure multicasting is a network technology used to achieve secure group communication (Trappe et al. 2001:1449). These two terms are often used interchangeably.

Secure group communication technologies mainly make use of encryption to keep the contents of a message secret (Rafaeli & Hutchison 2003:309). Members of the group share a common session key, i.e. a group key. Groups are dynamic and members can leave or join the group at any time. However, each time a change in the group occurs, the group key has to

change (Wong, Gouda & Lam 2000:16) and has to be redistributed to each member of the group. A single membership event thus affects the entire group, a limitation defined by Mitra (1997:277) as the 1-affects- n failure. Proposed solutions include locally maintained subgroups with subgroup keys (Dondeti, Mukherjee & Samal 2000:1681) and hierarchical schemes that distribute keys via a distribution tree (Banerjee, Bhattacharjee & Kommareddy 2002:205).

The main difference between these secure group communication technologies and the proposed system is the objectives of the systems. Secure group communication is designed for members of a group to communicate privately with one another by denying non-members access to the communicated messages. Non-members can, however, observe who the messages are from and who the messages are addressed to. The proposed system, on the other hand, is designed for one sender to communicate to a group in secret, not only denying access to unauthorised persons, but also keeping the existence of the communication secret.

Another difference in objectives between the systems is that secure group communication is intended for two-way communication where a message can originate from any member of the group and is broadcasted to the other members. However, the intent of the proposed system is a once-off one-way communication where a designated sender communicates a message to multiple receivers.

Secure group communication has a fixed list of members and changes to the list result in changes to the group key. The proposed system, on the other hand, has an ad hoc receiver's list that can change with each message without any additional changes to the system. With each communication, a new list of websites is communicated to the intended receivers – even should they be the same receivers as for the previous message.

Secure group communication technologies thus do not share the same objectives as the proposed system. Confidentiality of the information is provided by secure group communication, but since encryption is used to achieve the confidentiality, the system suffers from legality issues should encryption not be legal in that specific country. Secure group communication technologies thus also do not fully comply with the requirements for a secure communication system. The identities of the sender and receivers are not concealed in secure group communication and the possession of encrypted information could be considered

suspicious. In certain models, such as the secure and anonymous multicast (SAM) communications model proposed by Weiler and Plattner (2001:401), the identities of the members of the group are concealed. However, this includes the identity of the sender which is also not desirable since a level of sender authentication could be necessary.

3. OVERVIEW OF THE MESSAGE DISTRIBUTION SYSTEM

The proposed message distribution system uses image steganography to embed secret information in images which are then posted on public domain websites. Although confidentiality can be achieved by simply hiding the secret information in a single image, the use of a public channel to communicate the image could make it more vulnerable to attack. Furthermore, should the website hosting the image not be available due to technical difficulties or due to a firewall blocking access to the website, the receivers would not be able to retrieve the message – a direct violation of one of the requirements of the system. The suggested solution is thus to divide the secret information into n pieces which are embedded in n images and posted on n websites as illustrated in Figure 8.1.

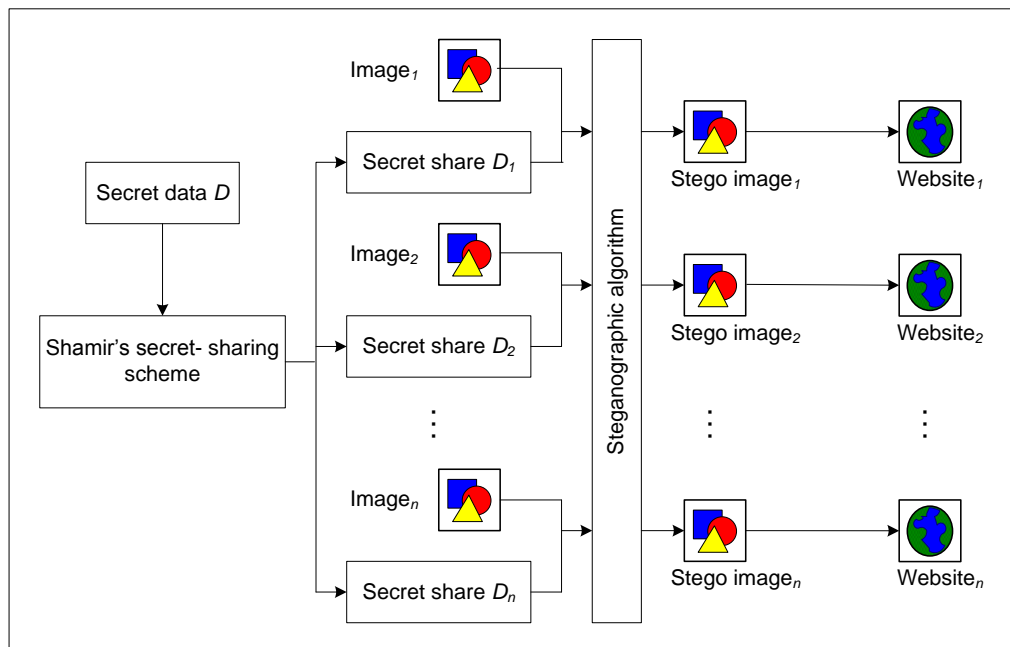


Figure 8.1. Process diagram of the message distribution system

Technical details of the message distribution system, specifically selection of the image steganography algorithm and cover images, as well as distribution of the stego images are discussed in section 5.

4. SECRET SHARING

This chapter proposes a system for secure one-to-many communication that uses Shamir's secret sharing scheme to divide the message into pieces which are then hidden in images using image steganography. In the message distribution system, the message acts as the secret to be shared. The next section discusses the principles of Shamir's secret sharing scheme. Section 4.2 examines related work where Shamir's secret sharing scheme was combined with image steganography and the implementation of Shamir's secret sharing scheme in the message distribution system is discussed in section 4.3.

4.1 Shamir's secret sharing scheme

Shamir (1979) proposed a scheme for sharing a secret, D , amongst n authorised parties. D is divided into n pieces D_1, \dots, D_n in such a way that

1. knowledge of any k or more D_i pieces makes D easily computable, and
2. knowledge of any $k - 1$ or fewer D_i pieces leaves D completely undetermined.

Such a scheme is called a (k, n) threshold scheme, since a secret is divided into n pieces in such a way that the secret can be reconstructed using k pieces, but not $k - 1$ pieces (Shamir 1979).

Threshold schemes were originally developed to solve the problem of a group of n mutually suspicious individuals having to share a secret (Shamir 1979). Since the individuals do not trust each other, each individual cannot have his own copy of the secret. The secret is divided into n pieces, called shares, and each individual, called a share holder, receives a piece of the secret. Should less than k share holders work together to try and access the secret they will be unsuccessful since the secret can only be decoded when k or more shares are combined.

Shamir's secret sharing scheme is based on polynomial interpolation (Shamir 1979). To divide the secret into n shares, a $k - 1$ polynomial is selected in which a_0 is the secret and the remaining $a_1 \dots a_{k-1}$ coefficients are randomly selected values:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{k-1} x^{k-1}$$

For each of the n shares, $f(x)$ is calculated for $x \in \{1, \dots, n\}$ and the resulting n $(x, f(x))$ value pairs are the shares that are given to the n share holders (Lin & Tsai 2004:405). Each share holder receives one $(x, f(x))$ share.

To recover the secret, at least k share holders have to combine their secret shares. A polynomial interpolation technique, such as the Lagrange method, is used to reconstruct the original $k - 1$ polynomial from the k $(x, f(x))$ value pairs (Lin & Tsai 2004:405). Once the original polynomial has been constructed, the secret again forms the a_0 coefficient of the polynomial and can be retrieved.

4.2 Related work on Shamir's secret sharing scheme

Most research done on combining secret sharing with images have focussed on sharing an image – a technology called secret image sharing. Naor and Shamir (1995) proposed a (k, n) threshold scheme to share image data. The original image is divided into random, noise-like images called shadows. The secret image is reconstructed by stacking the shadow images on one other. Further research on this scheme (Thien & Lin 2002:765; Chang et al. 2008:2433) optimises the generation and reconstruction of the shadow images.

In the field of steganography, research has been done to use steganography to hide the shadow images in cover images (Feng et al. 2005:327; Baek et al. 2010:325; Lin & Chan 2010:1887). Lin and Tsai (2004:405), for example, developed a system where an image is divided into n secret shares using Shamir's secret sharing scheme and then embedded in n cover images using image steganography.

In research that did not focus on secret image sharing, Potdar et al (2005:717) developed a system that uses Shamir's secret sharing scheme to again divide a secret into n pieces. However, the pieces are then embedded into different parts of the same cover image. This is done to ensure that the secret can still be recovered in the case of data loss due to image cropping.

These systems, however, were not developed for one-to-many communication systems and thus do not comply with the requirements of such a system.

4.3 Shamir's secret sharing scheme in the message distribution system

The message distribution system proposed in this chapter uses Shamir's secret sharing scheme to share a secret, but instead of sharing the secret between individuals, the secret is shared between locations. By hiding each piece of the secret in a different location, the confidentiality of the information is increased even with the presence of a public channel such as the Internet. Furthermore, since only k of the initial n shares are required, the secret can still be reconstructed even when some of the pieces are inaccessible due to an inaccessible website.

At the sender's side, the message is converted into a numerical representation if it is a text message. The conversion has to be reversible since the reconstructed numerical secret has to be converted back into a text message by the receivers. Each character in the text message is thus converted into its numerical ASCII representation between 0 and 255. To ensure that the message can be correctly converted back to a text message, the ASCII value of each character is stored in three digits and padded with leading zeros if less than three digits. The ASCII values are concatenated to form an integer.

To divide the integer value of the secret into shares, $k - 1$ coefficients, a_0, \dots, a_{k-1} , are created. The first coefficient a_0 , is assigned the value of the secret and the remaining coefficients are assigned randomly selected values. For each of the n shares, the $k - 1$ polynomial using the created coefficients is used to calculate $(x, f(x))$. The size of the randomly selected coefficients should be chosen carefully. If the coefficients have substantially smaller values than the secret, the addition done in the polynomial will not have a large enough effect on the number to change the message. For example, if the message 17/4UP was used to communicate perhaps a date and venue, the resulting integer value would be 49055047052085080. For $k = 5$, if $a_1 = 17$, $a_2 = 22$, $a_3 = 8$ and $a_4 = 13$, the value of $f(1)$ would be 49055047052085140 that is converted to 17/4Uî. The first share of the secret is thus almost identical to the actual secret with the exception of one character. Larger coefficients should thus be used so that the secret cannot be inferred from the secret shares.

Once the $n (x, f(x))$ value pairs have been created, the value pairs are embedded in stego images and distributed to websites.

At the receiver's side, k stego images are downloaded from websites and k value pairs are extracted from the stego images. Using Lagrange interpolation, the $k - 1$ polynomial is reconstructed and the secret is retrieved as the a_0 coefficient. The integer value of the secret is converted back into text by dividing the integer into groups of three digits and converting each three-digit-ASCII representation into a character representation.

An example of how the secret is divided into shares and the reconstruction of the secret is as follows:

Suppose that the secret message to be communicated is the abbreviation **SOS**. The ASCII values that represent the characters are 83 for S and 79 for O. The secret message converted into an integer value is thus 83079083. For this example, the secret is divided into four shares of which three should be combined to reconstruct the secret, thus $n = 4$ and $k = 3$. Using randomly selected values, -518542125 and 433514296 for the $k - 1$ coefficients, the resulting polynomial is:

$$f(x) = 83079083 - 518542125x + 433514296x^2$$

Using the polynomial, n secret share value pairs $(x, f(x))$ are calculated as follows:

$$x = 1: (1, -1948746)$$

$$x = 2: (2, 780052017)$$

$$x = 3: (3, 2429081372)$$

$$x = 4: (4, 4945139319)$$

Each of the value pairs is embedded in an image.

To reconstruct the message, three value pairs are extracted from any three of the stego images. Assuming that three value pairs were extracted, where $x = 1$, $x = 3$ and $x = 4$, value pairs are assigned as follows:

$$(x_0, y_0) = (1, -1948746)$$

$$(x_1, y_1) = (3, 2429081372)$$

$$(x_2, y_2) = (4, 4945139319)$$

The Lagrange formula:

$$l_i = \prod_{i \in k; j \in k, j \neq i} \frac{-x_j}{x_i - x_j}$$

Is used to create a set of three Lagrange basis polynomials:

$$l_0 = \frac{1}{6} x^2 - \frac{7}{6} x + 2$$

$$l_1 = -\frac{1}{2} x^2 + \frac{5}{2} x - 2$$

$$l_2 = \frac{1}{3} x^2 - \frac{4}{3} x + 1$$

Using the Lagrange basis polynomial, the following formula is used to create the original polynomial:

$$f(x) = \sum_{j \in k} y_j \cdot l_j(x)$$

The original polynomial is thus: $f(x) = 83079083 - 518542125x + 433514296x^2$

From this polynomial the a_0 coefficient is taken and converted into three digit groups: 83, 079 and 083 which are translated into SOS – the secret message.

5. TECHNICAL DETAILS OF THE MESSAGE DISTRIBUTION SYSTEM

Dividing the secret into shares amounts to one half of the message distribution system. The other half involves embedding the shares in images and posting the stego images on websites to facilitate communication. The security of the proposed system depends largely on the choice of parameters, such as image steganography algorithm, images and websites. The next section discusses the selection of a suitable image steganography algorithm. Section 5.2 discusses aspects of selecting cover images and websites and section 5.3 makes recommendations on how information such as a list of the websites, should be communicated to the receivers.

5.1 Selecting an image steganography algorithm

The most important requirement for a suitable image steganography algorithm for the proposed system is that images remain inconspicuous. Since the stego images will be openly posted on public websites, it is important that the images do not attract unnecessary attention. To facilitate the inconspicuousness of the images, the image steganography algorithm should provide a high level of invisibility and should be applied to popular image formats that are commonly found on the Internet. JPEG images are the most popular image format used on the Internet because of its small compression size (Wang & Wang 2004:78).

Additionally, a suitable image steganography algorithm should be robust against image manipulation attacks. Images posted on public websites have a high probability of being manipulated intentionally or unintentionally by website administrators or firewalls. JPEG steganography is thus recommended because it was optimally designed for JPEG images and provides high levels of invisibility and robustness.

5.2 Selecting the images and the websites

Receivers should be kept in mind when choosing cover images, since it will be the task of the receivers to locate and extract the information from the images. The extraction process should thus be easy to use. To recover the secret, receivers need knowledge of the websites on which the stego images are posted, but since more than one image can be posted on a website, the receivers should also have knowledge of which images to use. Communicating not only the websites, but also the exact images to the receivers add unnecessary overhead to the system and makes it vulnerable should this communication be intercepted. Using themed recall as introduced in chapter 6 to use images based on a specific theme, is thus recommended. Images should also fit into the general feel and topic of the website in order not to attract attention.

Websites on the other hand can be chosen at random as long as it has the capability that users can upload their own images to the website. Public sites such as online auction sites or Facebook are thus suitable and offer the necessary functionality. Public websites pose a higher risk of image manipulation, but JPEG steganography is robust against image manipulation attacks and can even withstand changes to the format of the image (Provos &

Honeyman 2001). Furthermore, even if some of the embedded information were lost due to image manipulation, the secret sharing scheme provides for the secret information to be retrievable as long as k of the n pieces are available.

5.3 Distributing list of websites to receivers

In order for intended receivers to be able to locate the images, extract the pieces, and to recover the secret, receivers will need additional information on the location of the images and the image steganography algorithm. One option to communicate this information to the receivers is to embed the algorithm that is necessary for extracting the secret shares in an image and communicate the image to the receivers, similarly to the decryptor distribution discussed in chapter 7. A list of the websites will then have to be sent to the receivers separately. If legal, encryption can be used to encrypt the list of websites prior to communication. However, the presence of encrypted information could cause suspicion which does not comply with the requirement of inconspicuousness. The list should thus be communicated to the receivers in a simple manner, for example via e-mail. Although there is a risk of the list being eavesdropped, it can be argued that a list of websites without knowledge of what they represent is not worth much and is certainly less suspicious than encrypted information.

Alternatively, if the receivers all have access to a computer device that can be trusted, a list of websites shared between the sender and the receivers could be agreed upon prior to travelling. The image steganography algorithm along with a list of websites can then be stored on the device and accessed when needed. However, should the device be lost or stolen, an unauthorised person could get access to the system.

6. EVALUATING THE MESSAGE DISTRIBUTION SYSTEM

As stated in section 1, the requirements of a secure communication system are confidentiality, inconspicuousness, legality and ease of use, in addition to the requirements for a one-to-many secure communication system, namely availability and a level of receiver anonymity. Confidentiality is provided by hiding the secret information in images using image steganography, and since encryption is not used, the legality of the system is also ensured. Shamir's secret sharing scheme adds another layer of confidentiality to the system

since a potential attacker will need to locate and extract at least k shares in order to reconstruct the secret information.

Since information is hidden in images on public websites, the communication is inconspicuous as there is no direct communication between the sender and the receivers. There is also no exchange of encrypted information that could cause suspicion. Availability is also achieved by using websites, since websites can be accessed from almost anywhere using devices such as computers and mobile devices.

The system proves to be at least as easy to use as other existing group communication systems. Although information regarding the websites on which the secret shares are stored needs to be distributed, it is no different than keys having to be distributed in an encryption-based system. It is actually less arduous since the list of websites need not be re-distributed for every change to the receivers list.

Finally, the identity of the sender and receivers can be kept confidential to a certain extent due to large traffic volumes and the relative anonymity of the Internet. Should a sender post a stego image on a public domain website where anyone can add images, the identity of the sender is hidden from unauthorised persons. Receivers also maintain a certain level of anonymity which can be enhanced by anonymising software. The proposed system thus proves to be at least more anonymous than direct communication between two parties.

7. CONCLUSION

This chapter proposed a system that uses image steganography to hide pieces of a secret in images and post the images on the Internet. In this manner a message can be securely communicated from a single sender to multiple receivers while maintaining the confidentiality of the secret message.

This chapter concludes the discussion on possible applications of image steganography for the three secure communication categories listed in chapter 1. In the last three chapters image steganography was combined with other security technologies to enable secure self-communication, secure one-to-one communication and secure one-to-many communication. Since each of the image steganography applications for the different secure communication

categories complied with the requirements of a secure communication system, image steganography successfully replaced cryptography in each application.

One important security aspect that is offered by cryptography through integrity check functions, but is not ensured by one of the image steganography applications, is the integrity of communicated information. Currently, there is no mechanism for checking that embedded information has not been changed during communication. Since integrity is also important for secure communication, the next chapter discusses a system that uses image steganography recursively to enable integrity checks.

CHAPTER 9

RECURSIVE IMAGE STEGANOGRAPHY FOR DATA INTEGRITY

1. INTRODUCTION

The previous three chapters have discussed solutions to how secure communication can be achieved in different scenarios using image steganography. What remains to be addressed is verifying that the information received is the same as the information that was sent, in other words the integrity of the information. Although steganography can ensure the confidentiality of hidden information, the integrity of the information is by no means ensured by simply hiding it in another object. Due to the nature of the embedding process, the embedded message shares the same bits as the stego object. In image steganography, changes to the stego image such as image manipulation techniques can change the bitwise composition of the stego image, and therefore also of the embedded message.

During communication, image manipulation techniques can be applied to an image either unintentionally or intentionally. A firewall could resize the image as part of its protocol and therefore destroy the embedded information unintentionally. An active attacker could intentionally alter the stego image in an attempt to either destroy the embedded information or to create confusion by changing it. Changes to the stego image can either be significant where changes are made to a large portion of the image or seemingly insignificant where only a few bytes of the image are changed.

Significant changes to the stego image, for example by image manipulation techniques, are easier to detect than smaller, seemingly insignificant changes, since insignificant changes are mostly intentional with the intention of adjusting the hidden information in such a way that the information is still intact, but incorrect.

In research, techniques such as watermarks (Fridrich 1999:26) and hash functions (Wong 1998:455) exist for verifying image authentication and data integrity and are discussed later in the chapter. All of these techniques, however, disclose additional information about the communication and can act as evidence that communication is taking place. This defeats the

purpose of steganography and could constitute a successful attack, even if the contents of the embedded message were not compromised since the communication is not inconspicuous.

It is thus necessary to develop a system that detects unintentional as well as intentional changes to a stego image, without disclosing additional information. This chapter proposes using image steganography recursively to verify the data integrity of embedded information. The fact that the embedded information is bitwise dependent on the stego image is used to visually verify the integrity of extracted information. Recursive steganography is done by embedding the message in an image, which is then embedded in another image. The message and the two images are bitwise linked in such a way that one cannot change without affecting at least one of the others. Changes to one of these objects result in visible changes to the others as is shown later in the chapter.

The remainder of the chapter first examines how current techniques are used for data integrity and why these techniques are not suitable when combined with image steganography. Section 3 discusses the proposed system that makes use of recursive image steganography to detect modifications to the stego image. Section 4 discusses technical details of the implementation of the proposed system. Results obtained from experimenting with the recursive image steganography prototype are discussed in section 5.

2. RELATED TECHNOLOGIES

The principle of the proposed system is that, when hiding information in images, modifications to the image leads to modifications to the embedded information and vice versa. Research similar to the proposed system has been done to detect these modifications in an attempt to verify the authenticity of digital images. Fragile watermarks were designed to detect changes in pixel values and are destroyed when these changes occur. Walton (1995:18) proposed a technique that calculates checksums of the most significant bits of an image and embeds the checksum in the least significant bits of pseudo-randomly selected pixels.

However, image authentication allows small modifications since small modifications do not dramatically change the visual contents of the image. Semi-fragile watermarks were thus proposed that are more robust to pixel modifications and can detect more significant changes,

but not insignificant changes (Fei, Kundur & Kwong 2006:43; Lin, Podilchuk & Delp 2000:152). Closely related to semi-fragile watermarks, Fridrich (1998a:404; 1998b) proposed robust watermarks that divides an image into medium-size blocks and inserts a watermark into each block. If an image feature comparable to the size of the block is removed or added, the watermark for that block will be destroyed. However, typical image processing operations will alter the image more uniformly and not just a single block and thus may not be detected (Fridrich 1999:26).

Self-embedding (Fridrich & Goljan 1999) is another technique proposed for correcting possible changes to the image as well as detecting image features that have been added or removed. In self-embedding, the entire image is embedded within itself. However, due to the very large payload capacity needed to embed the same image in itself, low invisibility and poor quality of the extracted image do occur (Fridrich 1999:26).

The main difference between these watermarking techniques and the proposed system lies in the intent of the two approaches. Watermarking techniques used for tamper detection are not concerned with the security of an embedded message, but rather with the authenticity of the visual contents of the image. Watermarking is designed to detect, and possibly correct, changes to the visual features of an image, for example the removal of a person in a digital photograph. Watermarking techniques are not designed to detect image processing techniques nor seemingly small changes to the image, since these changes are not important for authentication purposes (Rey & Dugelay 2002:613). Therefore, watermarking techniques are not suitable for ensuring the integrity and secrecy of an embedded message.

The proposed system, on the other hand, is concerned with firstly hiding the existence of the embedded message and secondly detecting changes to the stego image during communication. Since the focus of this chapter is on the authenticity of the embedded message all changes, significant as well as insignificant, should be detected as such changes could result in the message being modified.

Steganography literature has proposed other techniques for checking the integrity of the embedded message. Most of these techniques (Venkatraman, Abraham & Paprzycki 2004:347; Potdar, Han & Chang 2005:717; Park et al. 2007:393) either make use of cyclic redundancy checks (CRCs) or hash functions – security techniques that are often used for

data integrity – to detect unintentional changes to embedded information during communication. Fragile watermarks, for example, also often make use of hash functions (Wong 1998:455). However, these techniques are not as effective when implemented with image steganography, since when an attacker intentionally changes the stego image, the attacker could also intentionally change the integrity codes – either the CRC code or the hash value – and replace the existing codes with new ones to correspond to the newly changed information.

Additionally, the inclusion of CRCs or hash functions could result in affirmation of the presence of embedded information. The probability that information extracted from a random image consists of bits followed by an integrity code of those bits is very small and acts as evidence that information is embedded in the image. This constitutes a successful steganographic attack since the presence of embedded information has been discovered.

Digital signatures are also often used for checking data integrity (Wong 1998:455; Friedman 1993:905; Sun & Chang 2005:480) and are also frequently used by watermarking schemes (Schneider & Chang 1996:227). Similar to asymmetric encryption, the sender's private and public key pair is used for encryption and decryption to form a signature. However, in applications that require sender anonymity, the identity of the sender should also be hidden by removing identity information from the communication. The ability to decrypt embedded information with a sender's public key reveals his signature as well as his identity. An attacker could thus infer additional information from the image, although it will not be possible to change the embedded message without destroying it. However, digital signatures can thus be implemented to check whether changes were made to the stego image – either intentionally or unintentionally – at the expense of sacrificing additional information about the communication.

Ultimately, CRCs, hash functions, digital signatures, and watermarking techniques all require that a value calculated at the sender's side, called an integrity code, be compared to another value from the receiver's side to determine if the information has been modified. To ensure the covertness of the communicated information, the integrity code should not be dependent on the embedded message, since if discovered, the use of an integrity code could expose the existence of the embedded message.

In the recursive image steganography system proposed in this chapter, the inner image functions as integrity code since the inner image is ultimately the means for detecting modifications to the stego image. The recursive image steganography system thus embeds the message in the integrity code. However, it is not necessary to compare the inner image, in other words the integrity code, to anything else to determine whether information has been altered, since the alterations are immediately visible.

3. THE RECURSIVE IMAGE STEGANOGRAPHY SYSTEM

The basic idea behind the recursive image steganography system is to use image steganography as a means of checking the integrity of communicated information. When the stego image arrives at the receiver end, the assumption is that the receiver has no prior knowledge of the image. If the image was thus modified during communication, the receiver will not necessarily realise it. The recursive image steganography system offers a way of quickly detecting changes made to digital images through a visual verification instead of complicated calculations.

Since any form of digital data can be embedded in an image – including another image – recursively embedding a message in not one, but two images enables modifications to be detected at an earlier stage. The bitwise link between these three objects is such that, should the stego image be altered during communication, and with it the embedded message, then the alterations are also visible in the inner image. Section 5 of this chapter offers proof of the visible changes made to the inner image through experimental results. Figure 9.1 illustrates the recursive image steganography system.

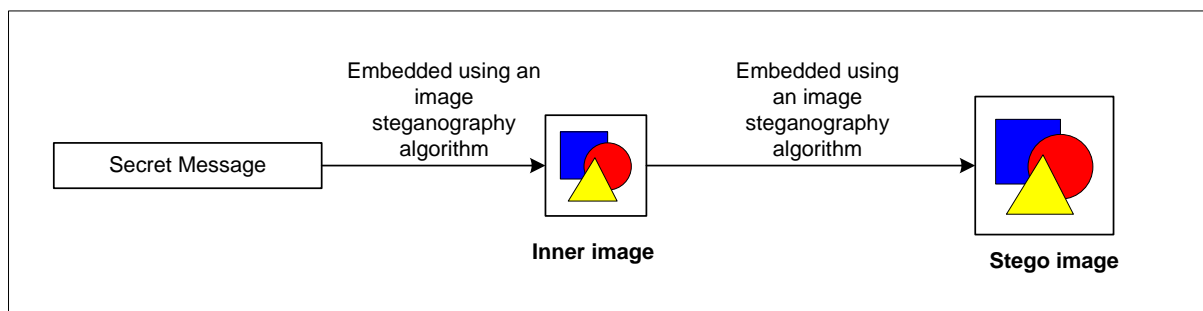


Figure 9.1. Recursive image steganography process

Unfortunately, the recursive image steganography system also introduces challenges that should be addressed. First of these is a technical challenge concerned with the implementation of the system: When the stego image is altered, the bitwise composition of the inner image also alters – a property that enables the proposed system to exist. However, these changes can often lead to a different image altogether. In fact, the changes to the inner image can be so severe that the image may not be recognised as a valid image anymore, thus a normal image viewer cannot display the image anymore.

The impact of this challenge can be debated: On the one hand it can be argued that if an inner image cannot be displayed, then the image must have been modified and the goal of detecting changes has been reached. On the other hand, if the goal is specifically to visually inspect the inner image for signs of modifications, then the system should be amended to display the inner image even if the image is no longer a valid image. This challenge and a solution for it are discussed in section 4.4.

The second challenge deals with the question of how much the receiver should know of the characteristics of the inner image. If the receiver has no knowledge of the inner image, the receiver will not know what to expect when visually checking the integrity of the data and may not detect modifications to the inner image. However, if the receiver knows exactly which image to expect, this information should be communicated by the sender at some time, creating a possible vulnerability in the system. The answer to this question lies in the choice of image to use as inner image as discussed in section 4.3.

4. DESIGN OF THE RECURSIVE IMAGE STEGANOGRAPHY SYSTEM

This section discusses technical details and recommendations made in designing a prototype for the recursive image steganography system. The first aspect that is examined in the next section is the choice of image steganography algorithm. Section 4.2 discusses the recursive image steganography process and recommendations are made regarding the choice of both cover image as well as inner image in section 4.3. Enabling the inner image to display even when changes to the stego image has affected the inner image's header is discussed in section 4.5. The implementation of the prototype is discussed in section 4.6.

4.1 Selecting an image steganography algorithm

The process of embedding the message in the inner image and the inner image in the stego image can be seen as two separate steganography applications and are treated separately. To embed the message in the inner image, a steganography algorithm that is not as robust against image manipulation attacks should be used. Algorithms that are robust against these attacks will allow for changes to be made to the stego image without affecting the embedded message and vice versa. Robustness is generally a desirable quality of an image steganography algorithm. However, when using the inner image as integrity code, it is important that changes to the embedded message are visible. For this reason (Johnson & Jajodia 1998(b):26), the prototype was implemented using LSB embedding to embed the secret message in the inner image.

Embedding the inner image in the stego image requires an image steganography algorithm with a high payload capacity since the stego image should be able to accommodate the inner image. LSB embedding was thus again chosen for the prototype due to the high payload capacity offered by the LSB embedding algorithm (Wu & Hwang 2007:1).

Using the same algorithm for both embedding processes has an advantage: If the receiver does not have access to a trusted computer device and cannot acquire a purpose built extractor without raising suspicion, then the receiver should be able to program the extractor. In this case, using the same algorithm would be beneficial.

However, when there is no boundary on computability and the receiver does have access to a trusted computer device on which a purpose built extractor has been stored, the receiver does not need to know the inner workings of the embedding processes. Different algorithms can then be implemented more easily with the advantage of the security gained by combining two steganography algorithms. By combining the two algorithms their strengths are combined and it is harder for an attacker to extract information from the stego image. The choice of algorithms thus also depends on whether or not the intended receiver has access to a trusted computer device or not.

4.2 Recursive image steganography

The main weakness of the LSB steganography algorithm is the ease in which it can be detected either statistically, or with brute force, should an attacker know that the technique was being used. One way of working around this weakness is to not use consecutive bytes for embedding the secret message, as would normally be done, but for the sender and receiver to share a secret key that specifies only certain pixels to be changed (Anderson & Petitcolas 1998:474).

However, including the use of a secret key in the recursive image steganography implementation adds unnecessary risk to the system since the key has to be communicated to the receiver. Ensuring that the key is securely communicated to the receiver shifts the focus from the confidentiality of the message to the confidentiality of the key.

In the implementation of the prototype, consecutive bytes of the stego image are thus used for LSB embedding without a stego key. However, for increased security the bits were inserted starting from the last byte of the image instead of the first. Since the information to be embedded in the stego image is not simply a small text message but another image, a high embedding rate of 3 bpp is required. The secret message is thus embedded in the LSB bit of every byte of the inner image, until the end of the message is reached. In turn, the LSB of every byte of the stego image is used to embed the inner image, until the end of the inner image is reached. An abstract representation of the distribution of the embedded message throughout the two images is given in Figure 9.2.

By replacing every 8th bit of the inner image data with a bit from the secret message and then every 8th bit of the stego image with a bit from the inner image, the embedded message is effectively distributed to every 64th bit of the stego image.

4.3 Image selection

A suitable cover image to use as stego image needs to be large enough to hide another image with an embedding rate of 3 bpp. If a 24-bit colour BMP image is used, the size of the stego image should thus be at least as many bytes as the number of bits that represents the inner image (not including the BMP header of the stego image).

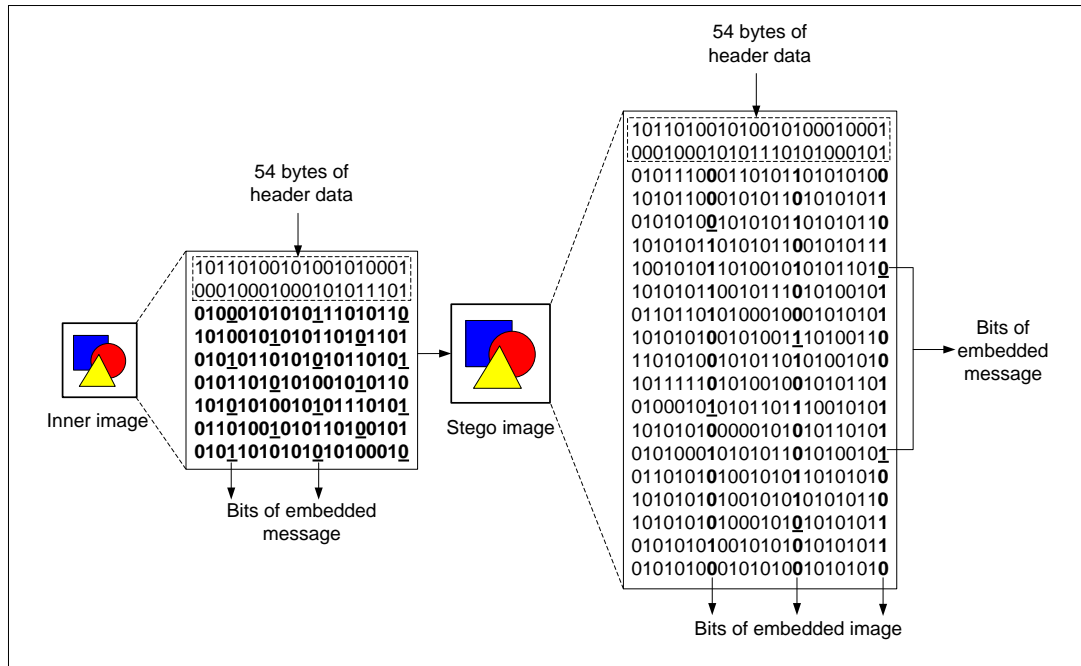


Figure 9.2. Representation of information distribution in inner image and stego image during recursive image steganography

The choice of inner image, however, is based on different requirements: When selecting an inner image, it is advisable to select a small, simple image. It should be just large enough for the secret message, but small enough to hide in another image. The image should be simple in the sense that it should be an image with few visual characteristics so that changes can easily be identified. Modifications to the stego image may only appear as noise in the inner image and if a too noisy image is chosen the modifications would blend in with the rest of the image.

Selecting a small, simple image as inner image also offers a solution to the question of how much the receiver needs to know about the inner image before checking for data integrity. If the inner image was a noisy image, then the receiver might not detect small changes to the image composition. However, if the image is a simple image such as a solid block of colour, then the receiver need not know the details of the inner image beforehand to be able to deduce that there is something wrong with the image. A few red pixels in a solid blue image, for example, can easily be detected visually. To further prove this point a simple image is used as inner image in the experiments with the prototype in section 5.

4.4 Displaying the inner image

When changes – especially significant changes – are made to the stego image, all the pixels of the stego image are modified, including those in which the 54 byte header of the inner image is embedded. Because a typical image viewer application first reads the BMP header before displaying the image, these changes could result in the inner image not being a valid BMP image anymore and the viewer will simply display an error message.

As argued earlier, this error message could be an indication of a breach of data integrity on its own. However, to investigate the visual impact of the changes, a method for displaying the image should be found.

The solution implemented in the prototype is to create a suitable BMP header and replace the damaged header with a new header before attempting to display the image. The remainder of the extracted image data remains unchanged so that unintentional or intentional changes made to the stego image during communication can still be seen. However, to create a suitable BMP header implies knowing the dimensions, colour depth, and compression of the inner image – knowledge that is not present anywhere in the stego image.

There are several approaches for the receiver to obtain this knowledge. One approach would be for the sender and receiver to exchange a suitable BMP header prior to the recursive steganography communication. By removing the BMP signature from the header data and by communicating the header as a bitstream, it would appear as seemingly random bits to an outside person. It would not be advisable for the entire image (prior to embedding) to be communicated to the receiver, since if an attacker were to acquire both images, one with and one without hidden information, the attacker could compare the images and find the presence of the embedded message.

The meta data stored in a BMP header is very general in the sense that two images that differ greatly visually could have the same header as long as their dimensions, colour depth, and compression are the same. Thus another approach is for the sender and receiver to communicate, not the exact inner image, but a different image with the same BMP header. Alternatively, the sender and the receiver can communicate, not the header data or an image, but rather the location of a suitable image in the public domain, for example a suitable image

on a website. The receiver can then successfully replace the header of the inner image with the header from the image from the website and display the image with an image viewer.

Finally, the prototype can be implemented with the BMP header data built into the system. This would imply that the dimensions, colour depth and compression of the inner image cannot change, but different images of the same size and BMP header can be used in future communication.

Although the process of acquiring the BMP header adds additional overhead to the system, the fact that the BMP header of the inner image can be replaced can also be used to our advantage. The main benefit of having to acquire and replace the BMP header of the inner image is that the BMP header can be removed from the inner image altogether. The only information to be embedded in the stego image then, is the image data. Not only does this reduce the size of the information that has to be embedded, but the absence of header data could also make it more difficult for an attacker to figure out the meaning of randomly extracted bits should an attacker be looking for evidence of embedded information.

4.5 The recursive image steganography prototype

The graphical user interface (GUI) of the recursive image steganography prototype was implemented using J# in Microsoft Visual Studio 2005. At the sender side the message is first embedded in the inner image before the inner image is embedded in the stego image. A screenshot of this recursive part of the system is given in Figure 9.3. The receiver uses the received image file as input to the prototype and extracts the information for visual inspection.

To test the functionality of the prototype as well as the effectiveness of the recursive image steganography system, results obtained from experimenting with the prototype are now discussed.

5. EXPERIMENTAL RESULTS

Changes that can be made to an image during communication have already been divided into significant changes and insignificant changes, depending on the amount of image data that is

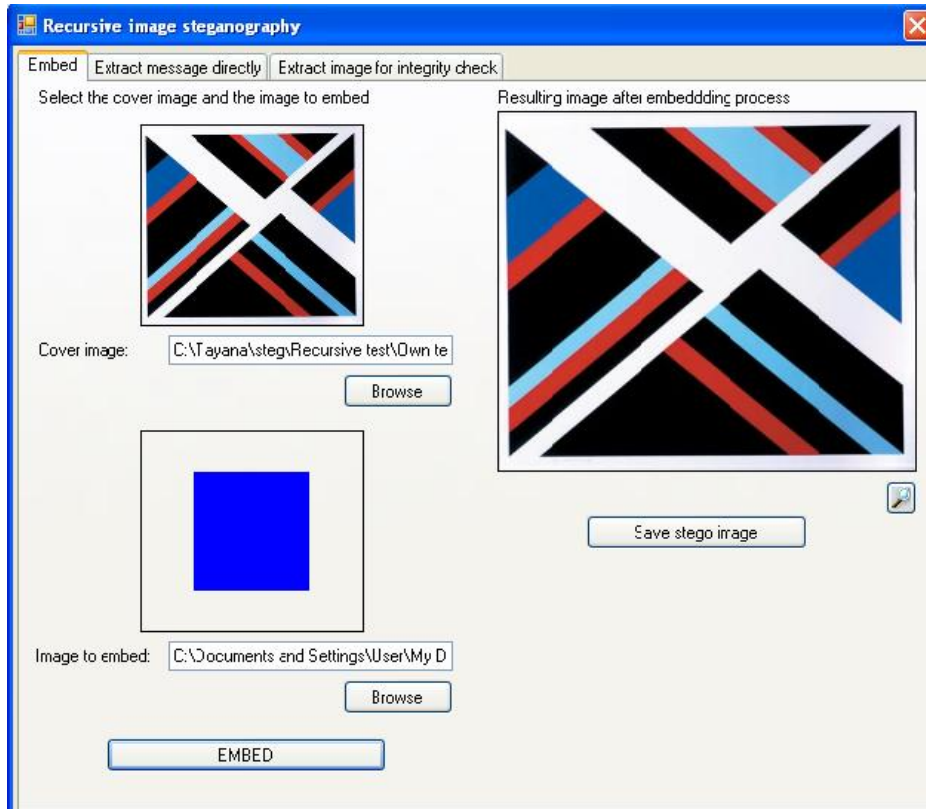


Figure 9.3. GUI of embedding phase of recursive image steganography prototype

modified. Significant changes lead to changes in large amounts or all of the stego image data, while insignificant changes are changes made to a couple of pixels of the stego image. The effect that these two different change categories have on the inner image as well as the embedded message differs since a different number of bytes are affected. The main reason for experimenting with the prototype is thus to examine the different effects of these changes and to test the effectiveness of the prototype in detecting significant and insignificant changes to the stego image.

Further reasons for experimenting with the prototype are to examine, both for significant as well as insignificant changes:

- (a) the extent to which changes made to the stego image are visible in the stego image itself,
- (b) the extent to which changes made to the stego image are visible in the inner image, and
- (c) the resulting effect of these changes on the embedded message.

For each experiment the 24-bit RGB BMP image shown in Figure 9.4 was used as stego image.

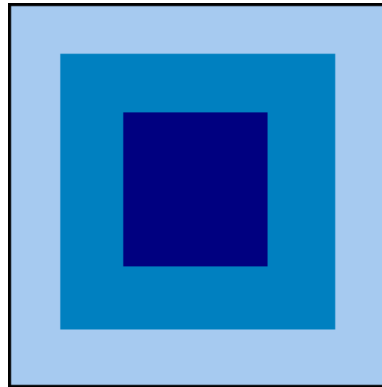


Figure 9.4. Stego image for experimental purposes

If the image was chosen as cover image in an image steganography application where the focus was exclusively on the invisibility of the embedded information, this image would not have been an appropriate choice since the image contains large areas of solid colours. However, for the purpose of the experiments in this section a “weak” image was chosen to more clearly display how changes affect the stego image. In noticing the sometimes tiny anomalies that occur in the stego image – anomalies that would not even have been visible if a noisier image was used – the importance of having a system that checks for changes is realised.

A simple, solid colour 24-bit BMP image, as seen in Figure 9.5, was used as inner image so that changes can be detected easily.



Figure 9.5. Inner image for experimental purposes

Experiments with significant and insignificant changes were done separately. The results of significant changes are discussed in section 5.1 and the results in insignificant changes are discussed in section 5.2.

5.1 Significant changes

To test the effect of significant changes on the recursive image steganography system, the prototype was used to embed a secret message in the inner image and in turn in the stego image. Significant changes were then made to the stego image to test the amount of tampering that the system will allow before the changes become visible. The resulting stego image, inner image and extracted message were documented and are shown in Table 9.1.

Table 9.1. Experimental results of significant changes and image manipulation techniques performed on stego image

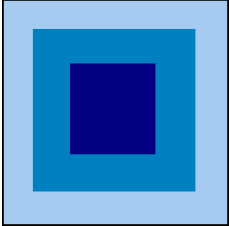
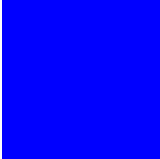
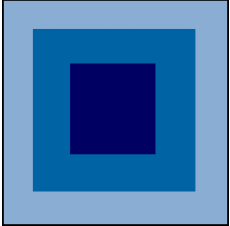
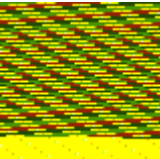
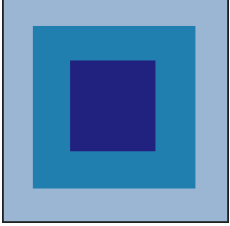
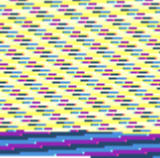
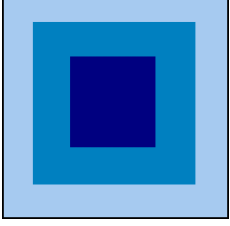
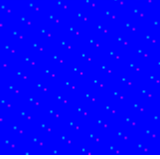
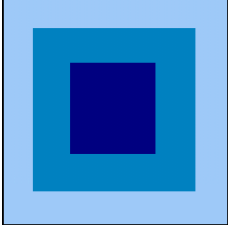
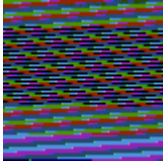
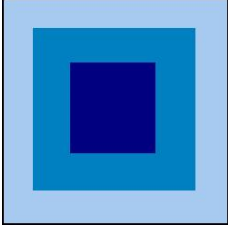
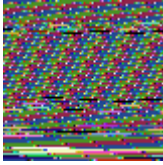
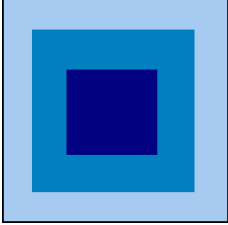
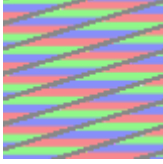
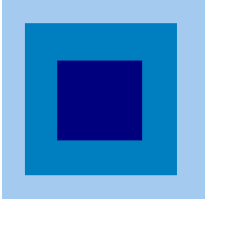
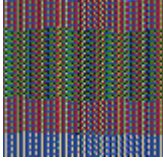
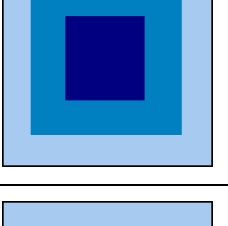
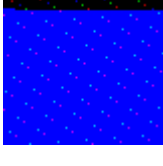
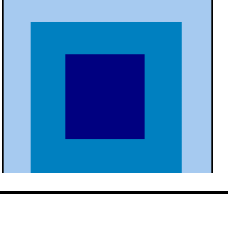
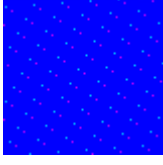
Image manipulation technique	Stego image	Extracted inner image	Extracted message
None			If you can read this message you are either the intended recipient or an excellent steganalyst
Brightness adjusted			\$I??I?
Contrast adjusted			K<_ik'MUA@:_*w%EG
Colour hue adjusted			If you can read this message you are either the intended recipient or an excellent steganalyst.

Table 9.1. Experimental results of significant changes and image manipulation techniques performed on stego image (continued)

Image manipulation technique	Stego image	Extracted inner image	Extracted message
Colour saturation adjusted			\$?IS?)\$?IS?IS\$?- \$?IS?IS\$LJ?IS?IS ?I
Image converted to JPEG and back to BMP			?IS?I?I?I\$?IS\$JR UjIS?IS?I)K1\$?IS ?IS,[&?IS?IS?I?I? IS?IS?JRUjIS?IS? D)K1\$?IS?IS,[&?IS?
Image rotated			\$?IS?IS?H?
Image cropped at both sides, top and bottom			r9?I,?G#?r9μ?K# ?r9?G\$?+?\$?bX\$?R)μ?E"?II\$μ?E"?HR RH?Ah4??9 E"?HR)\$?B?HR)μ?A
Image cropped at top			[blank message]
Image cropped at bottom			If you can read this message you are either the intended recipient or an excellent steganalyst.

The effects of the different image manipulation attacks are not always immediately obvious when looking at the stego image. Without knowledge of the image, the receiver will not be

able to detect that a significant change has been made to the stego image. However, the changes are visible in all of the inner images. Using this simple solid colour image as inner image, it is evident that, even if nothing was known of the visual characteristics of the image, the additional noise created by the significant change would still allow the receiver to detect changes. It is thus not crucial for the receiver to know the exact specifications of the inner image when using a simple image.

The most interesting aspect resulting from the experiments is that not all image manipulation techniques result in changes to the embedded message. The extracted message from the hue adjustment and the image cropping was intact and unchanged. This indicates that not all image manipulation techniques destroy embedded information, although the techniques are all detectable using the recursive image steganography prototype. This includes both attacks that are unintentional as well as attacks that are intentional.

Significant changes, however, are mostly applied uniformly to the entire stego image – and with it the entire inner image – and not just to parts of the stego image. It is thus a certainty that significant changes will affect the inner image and that the changes can be detected using recursive image steganography. Whether small changes that are made to certain parts of the stego image will also be visible, is examined next.

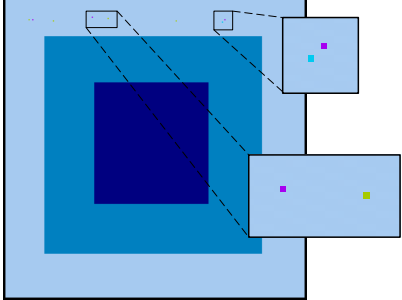
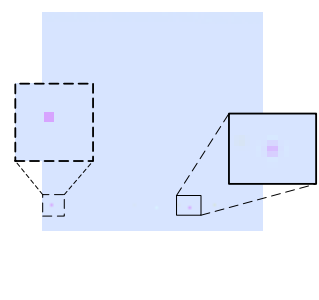
5.2 Insignificant changes

In an intentional image manipulation attack, an attacker may try to make only small, seemingly insignificant changes to parts of the stego image in an attempt to change the contents of the embedded message. To test the effectiveness of such an attack, experiments were done with insignificant changes starting with random changes to the stego image. For these experiments, the same stego image as for the experiments with significant changes were used. However, an image with the same specifications but in a lighter shade of blue was used as inner image to increase visibility.

The first experiment makes an insignificant change to the stego image in the form of changing a small number of randomly selected bytes. The goal of this experiment is not to change the message to something that the attacker wants, but rather just to randomly change the message without destroying it. Since the ASCII value of each character is stored in eight

bits, eight individual random bytes of the stego image were changed to different values. The resulting stego image, inner image and extracted message are shown in Table 9.2.

Table 9.2. Experimental results from altering eight individual, randomly selected bytes of the stego image

Stego image	Extracted inner image	Extracted message
		<p>If you can read this message you are either the intended recipient or an excellent steganalyst.</p>

Although small, the changes made to the colour values of the random pixels are visible as tiny pixel anomalies in the stego image. If a noisier stego image was used, these changes would not have been as noticeable in the stego image and without recursive image steganography they would probably not have been detected. With recursive image steganography, however, a small, but noticeable change also occurs in the inner image. Therefore, the change is detected. However, the embedded message remains unchanged.

When eight individual bytes are chosen at random, the probability of selecting a byte that has data from the message embedded in it is extremely slim. However, if an attacker were to change eight consecutive bytes of the stego image, the probability of selecting an occupied byte increases since every 64th bit of the stego image can contain embedded information. Table 9.3 shows the results of an experiment where eight consecutive bytes at a random location in the stego image are changed.

Although the changes to the stego image are visible not only in the stego image itself, but also in the inner image, there is still no change to the embedded message. The reason for this is that, in this particular experiment, the embedded message is not distributed across the entire stego image. The 95 bytes of the message is only embedded in the least significant bits of 760 bytes of the inner image. Since the total image size of the inner image is 18.8 KB, this surmounts to the embedded message occupying 3.96 % of the inner image data, excluding the

Table 9.3. Experimental results from altering eight consecutive bytes of the stego image

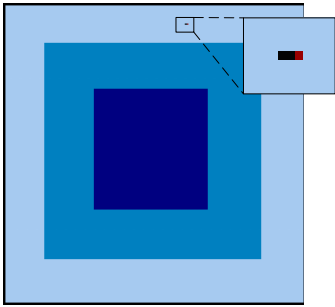
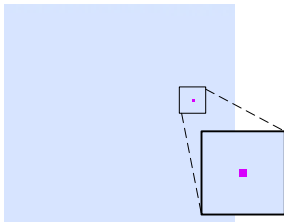
Stego image	Extracted inner image	Extracted message
		<p>If you can read this message you are either the intended recipient or an excellent steganalyst.</p>

image header. Thus only 0.4 % of the 183 KB of the stego image is used for the embedded message. At the maximum, the embedded message can constitute 1.5626 % of the stego image.

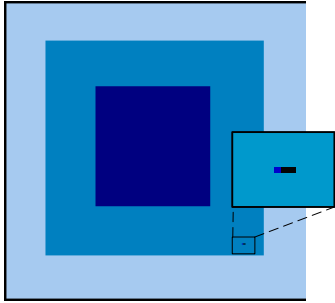
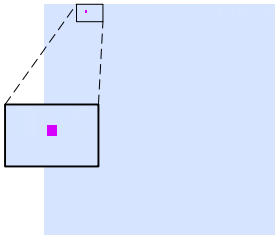
The size of the chosen stego and inner images are thus also important in the security of the steganography implementation. On the one hand, a smaller inner image is generally desired since it would lead to a smaller stego image, thus decreasing the suspicion caused by communicating large image files that are out of the ordinary. On the other hand, if both images are just large enough to contain the embedded message, the message could more easily be modified by an intentional, insignificant change since the message is distributed more uniformly across the entire image.

From the previous two experiments it was seen that the recursive steganography implementation is effective in detecting small insignificant changes and the effect of these changes were visible in the stego image as well as the inner image. However, the effect was not yet visible in the embedded message. It is thus still intriguing to investigate the scenario where the embedded message can be changed through small changes to the stego image. An experiment that attempts to achieve this scenario was done next.

To change the precise bytes of a stego image that will affect the embedded message requires intensive knowledge of the image steganography algorithm used, the size of the inner image, as well as the size of the embedded message. Selecting random locations for altering the bytes of the stego image is thus obviously not a feasible way of changing the embedded message. Since the message and inner image was embedded using the recursive image

steganography prototype, information on the image steganography algorithm and the sizes of the inner image and message are known. A final experiment was thus done by changing eight consecutive bytes of the stego image at the precise calculated location where the embedded message is located. The results of this experiment are shown in Table 9.4.

Table 9.4. Experimental results from altering eight consecutive bytes of the stego image at the precise location of the embedded message

Stego image	Extracted inner image	Extracted message
		<p>If you can read this message you are either the intended recipient or an excellent steganalyst.</p>

With intensive knowledge of the inner workings of the recursive image steganography prototype, it is thus possible to make small, seemingly insignificant changes to the stego image that will result in minor changes to the embedded message. These changes can, however, still be detected in both the stego image as well as the inner image.

When a textual message is communicated, like the one in the experiment, the receiver can detect that the information was modified, and can easily infer what the message was supposed to say. However, for increased security, an image steganography algorithm that offers more resistance against attacks may be needed. These algorithms often have a lower payload capacity in which case it is necessary to communicate only short, seemingly cryptic messages such as GPS coordinates, a time or a telephone number. However, modifications of one digit of these messages would alter its meaning entirely. It is then even more important to detect insignificant changes.

6. CONCLUSION

To ensure that communication is secure, communication has to be inconspicuous, invisible, legal and easy to use as specified in chapter 1. However, this chapter has shown that

embedded information can easily be changed during communication even in a communication system that complies with all of these requirements. This chapter thus proposed a recursive image steganography system that uses image steganography as a way of visually checking the integrity of received communication.

The results of experiments that were done with the recursive image steganography prototype showed that the system is successful in detecting changes made to the stego image. Significant changes to the stego image resulted in a distorted inner image and often completely changed the embedded message. Insignificant changes, however, did not change the embedded message except when the exact bytes containing the embedded message were calculated and changed. It is thus not easy for an attacker to make one or two changes to the bits of a message without intimate knowledge of the functionality of the recursive image steganography system.

CHAPTER 10

CONCLUSIONS

This chapter provides a summary of this dissertation in section 1 and provides ideas for future research in section 2.

1. SUMMARY

To communicate in secret is a necessity shared by many humans at some stage. For some the need for secure communication stems from a concern over privacy. For others the need for security is a result of the sensitive nature of the communicated information. Unauthorised access to secret information can have serious repercussions such as financial loss, breach of security or worse.

Over the years, many techniques were developed to ensure the confidentiality of communicated information, with cryptography techniques being the most popular. Communication that is encrypted, however, can occasionally give an attacker enough information to provoke a full-scale attack and steganography is often needed to hide the existence of communicated information. When comparing the security services offered by steganography and cryptography, it is clear that, on its own, steganography can not replace cryptography in all applications. For example, when implemented on its own, steganography simply can not offer the same level of authentication as offered by digital signature schemes. However, the secure communication applications researched in this dissertation successfully replaced encryption with image steganography. Image steganography can thus be used as an alternative to cryptography for secure communication.

For a communication system to be considered secure, the communication has to be inconspicuous, confidential, legal and easy to use. For each of the three secure communication categories, namely self-communication, one-to-one communication and one-to-many communication, this dissertation provided an image steganography application that complied with all of these requirements. The main objective of the dissertation as specified in chapter 1 was thus reached.

For self-communication, image steganography was combined with one-time passwords to provide a communication system that enabled the secure transport and storage of passwords. For one-to-one communication, image steganography was used to facilitate the communication of encrypted information, although in a manner in which the encrypted communication was hidden. In the same application, image steganography was also used to combine necessary tools and information in one image so that the receiver need only limited knowledge of the system to be able to receive the secret message. In one-to-many communication, image steganography was combined with Shamir's secret sharing scheme to divide a secret message into pieces and securely communicate the pieces from one sender to multiple receivers. The security of image steganography can thus easily be complemented by combining steganography with other security technologies.

2. FUTURE RESEARCH

Some ideas for future research include:

- **Application of steganography to offer non-repudiation**

In this dissertation, steganography applications were discussed that provided confidentiality, authentication, integrity, and availability. An image steganography application that provides non-repudiation would enable information to be embedded in an image in such a way that the person responsible for embedding cannot claim that the information was embedded by someone else.

- **Encryption algorithms for decryptor distribution**

For the decryptor distribution system implemented in chapter 7, the simple Caesar cipher was used as example of an encryption algorithm. Modern encryption algorithms, however, are more secure. For an alternative encryption algorithm to be suitable for the decryptor distribution system, the algorithm should be stored in a small enough file size to embed in an image.

- **Distribution of steganography algorithms and information to receivers**

For each of the image steganography applications discussed in chapters 6 to 9, the receivers could not extract the information without some knowledge of how the information is embedded as well as additional knowledge, for example the type of file that was embedded. For each of these applications, this dissertation attempted to provide solutions for how the information can be distributed to the receivers while maintaining the overall level of security. Enhanced methods for distributing the information to the receivers can however be investigated.

BIBLIOGRAPHY

Ahsan, K. & Kundur, D. 2002. Practical data hiding in TCP/IP. Proceedings of ACM Workshop on Multimedia Security.

Aloul, F., Zahidi, S. & El-Hajj, W. 2009. Two factor authentication using mobile phones. Proceedings of IEEE International Conference on Computer Systems and Applications, 641-644.

Anderson, R.J. & Petitcolas, F.A.P. 1998. On the limits of steganography. IEEE Journal on Selected Areas in Communication, 16(4):474-481.

Aparna, R. & Amberker, B.B. 2007. Dynamic authenticated secure group communication. World Academy of Science, Engineering and Technology Journal, 34:359-362.

Atoum, M.S. et al. 2011. A steganography method based on hiding secret data in MPEG/Audio layer III. International Journal of Computer Science and Network Security, 11(5):184-188.

Artz, D. 2001. Digital steganography: Hiding data within data. IEEE Internet Computing Journal, 5(3):75-80.

Baek, J. et al. 2010. (N, 1) secret sharing approach based on steganography with gray digital images. Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security, 325-329.

Bandyopadhyay, S.K. et al. 2008. A tutorial review on steganography. Proceedings of the International Conference on Contemporary Computing, 105-114.

Banerjee, S., Bhattacharjee, B. & Kommareddy, C. 2002. Scalable application layer multicast. Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, 205-217.

Bellare, M. & Rogaway, P. 1994. Entity authentication and key distribution. Lecture Notes in Computer Science, 773:232-249.

Bender, W. et al. 1996. Techniques for data hiding. IBM Systems Journal, 35(3):313-336.

Bloisi, D. & Iocchi, L. 2007. Image based steganography and cryptography. Proceedings of the International Conference on Computer Vision Theory, 127-134.

Bogdanov, A., Khovratovich, D. & Rechberger, C. 2011. Biclique cryptanalysis of the full AES, Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, 344-371.

- Boothe, R.G. 2002. Perception of the visual environment. Springer-Verlag Publishers.
- Borisov, N., Goldberg, I. & Brewer, E. 2004. Off-the-record Communication, or, Why not to use PGP? Proceedings of the ACM workshop on Privacy in the Electronic Society, 77-84.
- Chandramouli, R., Kharrazi, M. & Memon, N. 2004. Image steganography and steganalysis: Concepts and practice. Lecture Notes in Computer Science, 2939:204-211.
- Chang, C.C. et al. 2008. A novel secret image sharing scheme in color images using small shadow images. Information Sciences, 178(11):2433-2447.
- Chapman, D.B. & Zwicky, E.D. 1995. Building internet firewalls. O'Reilly Publishers.
- Choi, M. & Thang, N.M. 2010. An extensible authentication protocol with transport layer security and one time password in the multi hop mesh network. Recent Researches in Business Administration, Finance and Product Management, 88-93.
- Cifuentes, C. & Gough, K.J. 1995. Decompilation of binary programs. Journal of Software - Practice and Experience, 25(7):811-829.
- Compact Oxford English Dictionary of Current English. 2005. 3rd edition. Oxford University Press.
- Conklin, A. et al. 2004. Principles of computer security. McGraw-Hill Technology Education.
- Croft, N.J. & Olivier, M.S. 2005. Using an approximated one-time pad to secure short messaging service (SMS). Proceedings of the Southern African Telecommunications Networks and Applications Conference, 1:71-76.
- Currie, D.L. & Irvine, C.E. 1996. Surmounting the effects of lossy compression on steganography. Proceedings of the National Information System Security Conference, 194-201.
- Dajani, K., Owor, R. & Okonkwo, Z. 2010. The relevance of quantum cryptography in modern networking systems. Journal on Neural, Parallel and Scientific Computations, 18:391-400.
- Dondeti, L.R., Mukherjee, S. & Samal, A. 2000. Scalable secure one-to-many group communication using dual encryption. Journal of Computer Communications, 23(17):1681-1701.
- Dunbar, B. 2002. A detailed look at steganographic techniques and the use in an open-systems environment. SANS Institute, Information Security Reading Room, 1-11.

Elbouxhari, M., Azizi, M. & Azizi, A. 2010. Quantum key distribution protocols: A survey. *International Journal of Universal Computer Sciences*, 1(2):59-67.

Fei, C., Kundur, D. & Kwong, R.H. 2006. Analysis and design of secure watermark-based authentication systems. *IEEE Transactions on Information Forensics and Security*, 1(1):43-55.

Feng, J.B. et al. 2005. A new multi-secret images sharing scheme using Lagrange's interpolation. *Journal of Systems and Software*, 76(3):327-339.

Florêncio, D. & Herley, C. 2008. One-time password access to any server without changing the server. *Lecture Notes in Computer Science*, 5222:401-420.

Foley, J.D. et al. 1994. *Introduction to computer graphics*. Addison-Wesley Publishers.

Fridrich, J. 1998(a). Image watermarking for tamper detection. *Proceedings of the International Conference on Image Processing*, 2:404-408.

Fridrich, J. 1998(b). Methods for detecting changes in digital images. *Proceedings of the IEEE International Workshop on Intelligent Signal Processing and Communication Systems*.

Fridrich, J. 1999. Methods for tamper detection in digital images. *Proceedings of AMC Multimedia and Security Workshop*, 26-30.

Fridrich, J. 2010. *Steganography in digital media: Principles, algorithms and applications*. Cambridge University Press.

Fridrich, J. & Du, R. 1999. Secure steganographic methods for palette images. *Lecture Notes in Computer Science*, 1768:47-60.

Fridrich, J. & Goljan, M. 1999. Protection of digital images using self-embedding. *Proceedings of the Symposium on Content Security and Data Hiding in Digital Media*.

Fridrich, J., Goljan, M. & Du, R. 2001. Steganalysis based on JPEG compatibility. *Special Issue on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications*, 275-280.

Fridrich, J., Soukal, D. & Goljan, M. 2005. Maximum likelihood estimation of length of secret message embedded using PMK steganography in spatial domain. *Proceedings of IST/SPIE Electronic Imaging: Security, Steganography and Watermarking of Multimedia Contents*, 5681:595-606.

Friedman, G.L. 1993. The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905-910.

Gilbert, H. & Peyrin, T. 2010. Super-Sbox cryptanalysis: Improved attacks for AES-like permutations, *Proceedings of the International Workshop on Fast Software Encryption*, 365-383.

Gisin, N. et al. 2002. Quantum cryptography. *Reviews of Modern Physics*, 74:145-196.

Gollmann, D. 1999. *Computer security*. John Wiley & Sons Publishers.

Grandison, T. & Sloman, M. 2002. Specifying and analysing trust for internet applications. In: Monteiro, J.L., Swatman, P.M.C. & Tavares, L.V. *Towards the knowledge society*. SpringerLink, 145-157.

Grodzinsky, F.S., Miller, K. & Wolf, M.J. 2007. The ethical implications of the messenger's haircut: Steganography in the digital age. In: Himma, K.E. (ed.) *Internet Security: Hacking, Counter Hacking and Society*. Jones & Bartlett Publishers, 205-221.

Halevi, S. & Krawczyk, H. 1999. Public-key cryptography and password protocols. *ACM Transactions on Information and System Security*, 2(3):230-268.

Haller, N.M. 1994. The S/Key one-time password system. *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, 151-157.

Handel, T. & Sandford, M. 1996. Hiding data in the OSI network model. *Lecture Notes in Computer Science*, 1174:23-38.

ISO 7498-2. 1989. *Information processing systems: Open Systems Interconnection – Basic Reference Model, Part 2: Security Architecture*.

Jallad, K., Katz, J. & Schneier, B. 2002. Implementation of chosen-ciphertext attacks against PGP and GnuPG. *Lecture notes in Computer Science*, 2433:90-101.

Jamil, T. 1999. Steganography: The art of hiding information in plain sight. *IEEE Potentials*, 18(1):10-12.

Jefferson, D. et al. 2004. Analyzing internet voting security. *Communications of the ACM*, 47(10):59-64.

Jeong, J., Chung, M.Y. & Choo H. 2008. Integrated OTP-based used authentication scheme using smart cards in home networks. *Proceedings of the International Conference on System Sciences*, 294-301.

Johnson, N.F., Duric, Z. & Jajodia, S. 2001. Information hiding: Steganography and watermarking - attacks and countermeasures. Kluwer Academic Publishers.

Johnson, N.F. & Jajodia, S. 1998(a). Steganalysis of images created using current steganography software. *Lecture Notes in Computer Science*, 1525:273-289.

Johnson, N.F. & Jajodia, S. 1998(b). Exploring steganography: Seeing the unseen. *IEEE Computer Journal*, 31(2):26-35.

Johnson, N.F. & Jajodia, S. 1998(c). Steganalysis: The investigation of hidden information. *Proceedings of the IEEE Information Technology Conference*, 113-116.

Jorns, O., Bessler, S. & Pailer, R. 2005. An efficient mechanism to ensure location privacy in telecom service applications. *Proceedings of IFIP Conference on Network Control and Engineering for QoS, Security and Mobility III*, 165:57-68.

Kartalopoulos, S.V. 2006. A primer on cryptography in communications. *IEEE Communications Magazine*, 4:146-151.

Katz, J. & Schneier, B. 2000. A chosen ciphertext attack against several e-mail encryption protocols. *Proceedings of the Conference on USENIX Security Symposium*, 9:18.

Katzenbeisser, S. & Petitcolas, F.A.P. 1999. Information hiding techniques for steganography and digital watermarking. Artech House Books.

Kawaguchi, E. & Eason, R.O. 1999. Principles and applications of BPCS Steganography. *Proceedings of SPIE International Society for Optical Engineering*, 3528:464-473.

Khalili, A., Katz, J. & Arbaugh, W.A. 2003. Toward secure key distribution in truly ad-hoc networks. *Proceedings of the Applications and the Internet Workshops*, 342-346.

Kipper, G. 2003. Investigator's guide to steganography. Auerbach Publishers.

Kovacich, G. & Jones, A. 2002. What infosec professionals should know about information warfare tactics by terrorists. *Computers & Security*, 21(1):35-41 & 21(2):113-119.

Krenn, R. 2004. Steganography and steganalysis. <http://www.krenn.nl/univ/cry/steg/article.pdf>, last accessed on 2012-04-12.

Krinn, J. 2000. Introduction to steganography. <http://rr.sans.org/covertchannels/steganography.php>, last accessed on 2009-08-17.

Lampert, L. 1981. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770-772.

Lentra, A.K. & Verheul, E.R. 2001. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255-293.

Li, B. et al. 2011. A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing*, 2(2):142-172.

Lin, P.Y. & Chan, C.S. 2010. Invertible secret image sharing with steganography. *Pattern Recognition Letters*, 31(13):1887-1893.

Lin, E.T., Podilchuk, C.I. & Delp, E.J. 2000. Detection of image alterations using semi-fragile watermarks. *Proceedings of SPIE Conference on Security and Watermarking of Multimedia Contents*, 3971:152-63.

Lin, C.C. & Tsai, W.H. 2004. Secret image sharing with steganography and authentication. *Journal of Systems and Software*, 73:405-414.

Lo, J.L., Bishop, J.M. & Eloff, J.H.P. 2008. SMSec: An end-to-end protocol for secure SMS. *Journal for Computers and Security*, 27(5):154-167.

Lou, D. & Liu, J. 2002. Steganographic method for secure communication. *Computer and Security*, 21(5):449-460.

Marvel, L.M., Boncelet, C.G. & Retter, C.T. 1999. Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8(8):1075-1083.

Mavrakis, N. 2003. Vulnerabilities of ISPs: an overall look. *IEEE Potentials*, October/November:9-15.

McDonald, D.L., Atkinson, R.J. & Metz, C. 1995. One time password in everything (OPIE): Experiences with building and using stronger authentication. *Proceedings of the USENIX UNIX Security Symposium*, 5:16.

Menezes, A., Van Oorschot, P. & Vanstone, S. 1996. *Handbook for applied cryptography*. CRC Press.

Mitra, S. 1997. Iolus: A framework for scalable secure multicasting. *Proceedings of ACM SIGCOMM*, 277-278.

Moerland, T. 2003. *Steganography and steganalysis*. Leiden Institute of Advanced Computing Science <http://www.liacs.nl/home/tmoerl/privtech.pdf>, last accessed on 2006-05-01.

Murray, J.D. & Van Ryper, W. 1996. Encyclopedia of graphics file formats. O'Reilly Publishers.

Naghsh-Nilchi, A.R. & Pourmohammadbagher, L. 2006. A new approach to steganography using sinc-convulsion method. International Journal of Applied Science, Engineering and Technology, 2(3):147-152.

Naor, M. & Shamir, A. 1995. Visual cryptography. Lecture Notes in Computer Science, 950:1-12.

Oprea, A. et al. 2004. Securing a remote terminal application with a mobile trusted device. Proceedings of the 20th Annual Computer Security Applications Conference, 438-447.

Orchard, M. & Bouman, C. 1991. Color quantization of images. IEEE Transactions on Signal Processing, 39(12):2677-2690.

Owens, M. 2002. A discussion of covert channels and steganography. SANS Institute, Information Security Reading Room.

Papapanagiotou, K. et al. 2005. Alternatives for multimedia messaging system steganography. Lecture Notes in Computer Science, 3802:589-596.

Park, Y. et al. 2007. A novel steganographic system with information integrity. IEICE Electronics Express, 4(12):393-399.

Petitcolas, F.A.P., Anderson, R.J. & Kuhn, M.G. 1999. Information hiding – A survey. Proceedings of the IEEE Special Issue on Protection of Multimedia Content, 87(7):1062-1078.

Pfitzmann, B. 1996. Information hiding terminology - Results of an informal plenary meeting and additional proposals. Lecture Notes in Computer Science, 1174:347-350.

Pfleeger, C.P. 2000. Security in computing. 2nd Edition. Prentice Hall.

Philjon, J.T.L. 2011. Metamorphic cryptography - A paradox between cryptography and steganography using dynamic encryption. Proceedings of the International Conference on Recent Trends in Information Technology, 217-222.

Por, L.Y., Ang, T.F. & Delina, B. 2008. WhiteSteg: A new scheme in information hiding using text steganography. WSEAS Journal of Transactions on Computers, 7(6):735-745.

Potdar, V.M., Han, S. & Chang, E. 2005. Fingerprinted secret sharing steganography for robustness against image cropping attacks. Proceedings of the IEEE International Conference on Industrial Informatics, 717-724.

Prakash, M.V., Infant, P.A. & Shobana, S.J. 2010. Eliminating vulnerable attacks using one-time password and passtext - Analytical study of blended schema. *Universal Journal of Computer Science and Engineering Technology*, 1(2):133-140.

Provos, N. 2001. Defending against statistical steganalysis. *Proceedings of the USENIX Security Symposium*, 323-335.

Provos, N. & Honeyman, P. 2001. Detecting steganographic content on the internet. *Proceedings of the Internet Society Symposium on Network and Distributed System Security*.

Rabah, K. 2004. Steganography – The art of hiding data. *Information Technology Journal*, 3(3):245-269.

Rafaeli, S. & Hutchison, D. 2003. A survey of key management for secure group communication. *ACM Journal of Computing Surveys*, 35(3):309-329.

Rey, C. & Dugelay, J. 2002. A survey of watermarking algorithms for image authentication. *EURASIP Journal on Applied Signal Processing*, 6:613-621.

RSA SecureID. www.emc.com/security/rsa-securid.htm, last accessed on 2012-04-12.

Salomon, D. 2004. *Data compression: The complete reference*. Springer-Verlag Publishers.

Sattarova, F.Y. & Tai-hoon, K. 2009. Review on YCrCb color space optimization. TV images compression, algorithm of signal sources isolation and optical fiber, *International Journal of Smart Home*, 3(4):43-62.

Schneider, M. & Chang, S. 1996. A robust content based digital signature for image authentication. *Proceedings of the International Conference on Image Processing*, 3:227-230.

Schneider, G.M. & Gersting, J.L. 2004. *Invitation to computer science*. Course Technology.

Schneier, B. 1996. *Secrets & Lies: Digital Security in a Networked World*. Wiley Computer Publishing.

Shamir, A. 1979. How to share a secret. *Communications of the ACM*, 22(11).

Shirali-Shahreza, M. 2006. Stealth steganography in SMS. *Proceedings of IFIP International Conference on Wireless and Optical Communications Networks*.

Shirali-Shahreza, M. 2008. Text steganography by changing words spelling. *Proceedings of the International Conference on Advanced Communication Technology*, 3:1912-1913.

Shirali-Shahreza, M. & Shirali-Shahreza, S. 2006. Collage steganography. Proceedings of the IEEE/ACIS International Conference on Computer and Information Science, 316-321.

Simmons, G.J. 1983. The prisoners' problem and the subliminal channel. Advances in Cryptology: Proceedings of CRYPTO, 51-67.

SolidPass. www.solidpass.com, last accessed on 2012-04-12.

Sun, Q. & Chang, S. 2005. A secure and robust digital signature scheme for JPEG2000 image authentication. IEEE Transactions on Multimedia, 7(3):480-494.

Syverson, P. 1994. A taxonomy of replay attacks. Proceedings of the Computer Security Foundations Workshop, 187-191.

Szaban, M. & Sredynski, F. 2012. Dynamic cellular automata-based B-boxes. Lecture Notes in Computer Science, 6927:184-191.

Thien, C.C. & Lin, J.C. 2002. Secret image sharing. Computers & Graphics, 26(5):765-770.

Trappe, W., Song, J., Poovendran, R. & Liu, K.J.R. 2001. Key distribution for secure multimedia multicasts via data embedding. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 3:1449-1452.

Tsai, C.R. 2003. Non-repudiation in practice. <http://dsns.csie.nctu.edu.tw/iwap/proceedings/sessionD/6.pdf>, last accessed on 2009-07-10.

Tzeng, C.H., Yang, Z.F. & Tsai, W.H. 2004. Adaptive data hiding in palette images by color ordering and mapping with security protection. IEEE Transactions on Communications, 52(5):791-800.

Valente, E., Redd, J. & Northcutt, S. 2009. Two-factor authentication: Can you choose the right one? SANS Institute, Information Security Reading Room.

Venkatraman, S., Abraham, A. & Paprzycki, M. 2004. Significance of steganography on data security. Proceedings of the International Conference on Information Technology: Coding and Computing, 2:347.

Walton, S. 1995. Information authentication for a slippery new age. Dr. Dobbs Journal, 20(4):18-26.

Wang, H. & Wang, S. 2004. Cyber warfare: Steganography vs. steganalysis. Communications of the ACM, 47(10):76-82.

Wang, X. & Yu, H. 2005. How to break MD5 and other hash functions, Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 1-18.

Web definition: secure. Webopedia, <http://webopedia.com/TERM/C/Communications.html>, last accessed on 2009-03-10.

Westfeld, A. 2001. F5 – A steganographic algorithm: High capacity despite better steganalysis. Lecture Notes in Computer Science, 2317:289-302.

Weiler, N. & Plattner, B. 2001. Secure and anonymous multicast framework. Proceedings of the IFIP Conference in Communication and Multimedia Security, 401-410.

Weiss, M. 2009. Principles of Steganography. <http://www.math.ucsd.edu/~crypto/Projects/MaxWeiss/steganography.pdf>, last accessed on 2012-04-12.

Whitman, M.E. & Mattord, H.M. 2003. Principles of information security, Thomson Course Technology Publishers.

Wiggins, R.H. et al. 2001. Image file formats: Past, present and future. RadioGraphics, 21(3):789-798.

Wong, P. 1998. A public key watermark for image verification and authentication. Proceedings of the International Conference on Image Processing, 1:455-9.

Wong, K., Cheung, C. & Po, L. 2002. Merged-color histogram for color image retrieval. Proceedings of the IEEE International Conference on Image Processing, 3:949-952.

Wong, C.K., Gouda, M. & Lam, S.S. 2000. Secure group communications using key graphs. IEEE/ACM Transactions on Networking, 8(1):16-30.

Wu, N.I. & Hwang, M.S. 2007. Data hiding: Current status and key issues. International Journal of Network Security, 4(1):1-9.

Wu, D.C. & Tsai, W.H. 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition Letters, 24(9):1613-1626.

Xi, L., Ping, X. & Zhang, T. 2010. Improved LSB matching steganography resisting histogram attacks. Proceedings of the IEEE International Conference on Computer Science and Information Technology, 203-206.

Xia, H. & Brustoloni, J.C. 2005. Hardening web browsers against man-in-the-middle and eavesdropping attacks. Proceedings of the International Conference on World Wide Web, 489-498.

Zhang, X. & Wang, S. 2005. Steganography using multiple-base notational system and human vision sensitivity. *IEEE Signal Processing Letters*, 12(1):67-70.

Zhou, Y. 2011. Image encryption using the image steganography concept and PLIP model. *Proceedings of the International Conference on System Science and Engineering*, 699-703.

APPENDIX A: PUBLICATIONS DERIVED FROM THIS WORK

Morkel, T., Eloff, J.H.P. & Olivier, M.S. 2006. Using image steganography for decryptor distribution. Proceedings of the On the Move to Meaningful Internet Systems workshop on Information Security, Lecture Notes in Computer Science 4277:322-330.

Morkel, T., Eloff, J.H.P., Olivier, M.S. & Venter, H.S. 2005. One time passwords in a mobile environment using steganography. Proceedings of IEEE conference on Pervasive Systems workshop on Security in Pervasive and Ubiquitous Computing.

Morkel, T., Eloff, J.H.P. & Olivier, M.S. 2005. An overview of image steganography. Proceedings of Information Security South Africa conference.