A Study of Recent Results on Provably Secure Steganography

A dissertation submitted in partial fulfillment of the requirements for the M.Tech (Computer Science) degree of the Indian Statistical Institute, Kolkata.

By

Chandrasekhara Rao Ch

under the supervision of

Dr. Subhamoy Maitra
Applied Statistics Unit

INDIAN STATISTICAL INSTITUTE
203, B.T. Road, Kolkata - 700035
21 July' 2004

Indian Statistical Institute,
203, Barrackpore Trunk Road,
Kolkata - 700 035

Certificate Of Approval

This is to certify that this thesis titled " A Study of Recent Results on Provably Secure Steganography " submitted by Chandrasekhara Rao Ch. towards partial fulfillment of requirements for the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata embodies the work done under my supervision.

Subhamoy Maitra
21/07/2004

Dr. Subhamoy Maitra,
Applied Statistics Unit,
Indian Statistical Institute,
Kolkata - 700 108.

[illegible] 23/9/08

ABHIK MUKHERJEE
CST Dept, BECDU

(External Expert)

Acknowledgement

I take pleasure in thanking Dr.Subhamoy Maitra for his friendly guidance throughout the dissertation period. His pleasant and encouraging words have always kept my spirits up.

I would also like to express my sincere gratitude to Mr. T.K.Das for agreeing to discussions and valuable suggessions. I would like to thank members of Cryptology Research Centre, ISI, Kolkata.

Finally I take the opportunity to thank my classmates, friends and family members for their encouragement to finish this work.

Chandrasekhara Rao Ch

Abstract

Steganography is the science of hiding the very existence of secret message within innocuous looking message. Cyptography is about to concealing the content of the message, steganography is about to concealing their very existence. Thought, very little work has been attempted to formalize steganography, and most of the literature consists of heuristic approaches. A few papers have given information theoretic models for steganography, but these models are limited in the same way that infromation theoretic cryptography is limited. In CRYTPO-2002 Nicholas J. Hopper , John Langford and Luis von Ahn gave the first theoretical concepts of steganography from Complexity-theory. But there is a big gap between the theoretical and practical aspects of the steganography from Complexity-theoretic point of view. In our work we done comprehensive study of steganography from Complexity-theoretic point of view and based on those ideas we propose an algorithm which is practical and can be implemented efficiently. And our algorithm is secure in presence of passive adversary and more over robust to some bounded active adversary.

Contents

1 Introduction **2**

1.1 Steganography: History 2

1.2 Simple systems 3

1.3 Public Key Steganography 4

1.3.1 In Presence of passive warden 4

1.3.2 With an active warden 4

2 Public-Key Encryption **6**

2.1 Definition of Public - Key Encryption 6

2.2 Simple Examples of PC: The Trapdoor Function Model 7

2.3 Problems with the Trapdoor Function Model 8

2.4 Problems with Deterministic Encryption in General 8

2.5 Probabilistic Public-Key Encryption 9

2.5.1 Encrypting Single Bits: Trapdoor Predicates 9

2.5.2 Encrypting Single Bits: Hard Core Bits 9

3 Provably Secure Steganography **11**

3.1 Basics 11

3.1.1 Notations from Cryptography 11

3.1.2 Steganography 12

3.1.3 Steganographic Secrecy 14

3.2 Stateless Steganographic Secrecy 15

3.3 Stateful Steganographic Secrecy 17

4 Provably secure Stego-Algorithm **21**

4.1 Efficient Probabilistic Encryption 21

4.2 Provably secure Stego-Algorithm. 22

5 Conclusion **25**

Chapter 1

Introduction

1.1 Steganography: History

Steganography is the science or possibly art of hiding the very existence of the secret message within a cover-text. Steganography concerns itself with ways of embedding a secret message (which might be a copyright mark, or a covert communication, or a serial number) in a cover message (such as a video film, an audio recording, or computer code). The embedding is typically parametrized by a key: without knowledge of this key (or a related one) it is difficult for a third party to detect or remove the embedded material. Once the cover object has material embedded in it, it is called a stego object. Thus, for example, we might embed a mark in a cover-text giving a stegotext; or embed a text in a cover image giving a stego-image: and so on[9].

While classical cryptography is about to concealing the content of messages. stegano graphy is about to concealing the very existence of messages, and it goes back to ancient times. Khan tells of a classical Chinese practice of embedding a code ideogram at a prearranged place in a dispatch[3]; of the warning the Greeks received of Xerxes' intentions from a message underneath the wax of a writing tablet: and a trick of dotting successive letters in a cover-text with secret ink, due to Aeneas the Tactician.The opponent may be passive. and merely observe the cover-text. but he may also active.In the US post office during the second world war, postal censors deleted lovers' X's, shifted watch hands, and replaced items such as loose stamps and blank paper. They also rephrased telegrams; in one case, a sensor changed 'father is dead' to 'father is deceased', which elicited the reply 'isfather dead or deceased?'

The study of this subject in the open scientific literature may traced to Simmons, who in 1983 formulated it as the prisoners' problem[11]; Alice and Bob are in jail. and wish to hatch an escape plan. All their communications pass through the warden, Ward. If Ward sees any encrypted messages, he will frustrate their plan by putting them into solitary confinement. So they must find some way of hiding their cipher-text in an innocuous looking cover-text. As in the related field of cryptography, we assume that the mechanism in use is known to warden, and so the security must rely solely on a secret key.

There are many real applications of steganography[6]. Apparently, during the 1980's, British Prime Minister Margaret Thatcher became so irritated at press leaks of cabinet documents that she had the word processors programmed to encode their identity in the word spacing of documents, so that disloyal ministers could traced. Similar techniques are now undergoing trails in an electronic publishing project, with a view to hiding copyright messages and serial numbers in documents[7]. Steganography must not be confused with cryptography, where we transform the message so as to make its meaning obscure to a person who intercepts it.Such protection is often not enough: the detection of encrypted message traffic between a soldier and a hostile government, or between a known drug-smuggler and someone not yet under suspicion, has obvious implications. we still have no comprehensive theory of steganography, in the way that Shannon gave us theory of encryption and Simmons of authentication.

1.2 Simple systems

A number of computer programs are available that will embed information in an image. Some of them just set the least significant bits of the image pixels to the bits of the embedded information[13]. Information embedded in this way may be invisible to the human eye but is trivial for an alert third party to detect and remove. Slightly better systems assume that both sender and receiver share a secret key and use a conventional cryptographic key-stream generator[10] to expand this into a long pseudo-random key-stream. The key-stream is then used to select pixels or sound samples in which the bits of the cipher-text are embedded.

Not every pixel may be suitable for encoding cipher-text: changes to pixels in large fields of monochrome color, or that lie on sharply defined boundaries, might be visible. So some systems have an algorithm that determines whether a candidate pixel can be used by checking that the variance in luminosity of the surrounding pixels is neither very high (as on a boundary) nor very low (as in a monochrome field). Wherever a pixel passes this test, we can tweak its least significant bit to embed a bit of our message. Such schemes can be destroyed in a number of ways by an opponent who can modify the stego-image. For example, almost any trivial filtering process will change the value of many of the least significant bits. One possible countermeasure is to use redundancy: either apply an error correcting code, or simply embed the mark a large number of times. For example, the "Patchwork" algorithm of 'Bender' hides a bit of data in an image by increasing the variance in luminosity of a large number of pseudo-randomly chosen pixel pairs; and a similar system was proposed by Pitas[6].

1.3 Public Key Steganography

Until recently, it was generally assumed that, in the presence of a capable motivated opponent, steganography required the pre-existence of a shared secret so that the two communicating parties could decide which bits to tweak.

1.3.1 In Presence of passive warden

In a workshop the authors, Ross J. Anderson, F. A. P. Petitcolas in their paper [1] showed that public-key steganography is possible in the presence of a passive warden. Given a cover-text in which any cipher-text at all can be embedded, then there will usually be a certain rate at which its bits can be tweaked without the warden noticing. So suppose that Alice can modify at least one out of every k bits of the cover-text. This means that Ward cannot distinguish the parity of each successive block of k bits from random noise, and it follows that Alice can encode an arbitrary pseudo-random string in these parities. This pseudo-random material lies in plain sight; anyone can read it. So Ward cannot tell the difference between stegotext and pure cover-text by randomness tests; a suitable parity check function will extract pseudo-random-looking data from any cover-text in which information could have been embedded at all.

Now suppose that Alice and Bob did not have the opportunity to agree a secret key before they were imprisoned, but that Bob has a public key that is known to Alice. She can take her covert message, encrypt it under his public key, and embed it as the parity of successive blocks. Each possible recipient will then simply try to decrypt every message he sees, and Bob alone will be successful. In practice, the value encrypted under a public key could be a control block consisting of a session key plus some padding, and the session key would drive a conventional steganographic scheme. Normal public key cryptography means that users can communicate confidentially in the absence of previously shared secrets; The intuitive construction of public key steganography shows that they can also communicate covertly.

1.3.2 With an active warden

The open question left in [1] was whether public key steganography was possible in the presence of a warden who is active rather than passive. The original construction fails in this case, as Ward can also tweak one bit in every k; he could even set the parity of each successive block to zero. some idea how concealed public key communication may still be possible in the presence of an active warden provided that the model of the Prisoners' Problem is changed slightly.

Assume that the stegomessages Alice sends to Bob will be sent to other recipients too, such as a mailing list or Usenet newsgroup. Also assume that Ward and Alice are each able to tweak at most one bit in k of the content (as above, Ward might exceed the limit's of Bob's rights if he distorts the communication channel to the point that it becomes unusable). Alice can choose a short one-time key that selects some permutation of the cover-text bits, and she hides a message as the parity of successive k-tuples of bits in this permuted sequence. Ward, suspecting that this method may be in use, alters 1 in k of the stegotext bits; this is the best he can do since he does not know what one-time key Alice used. This corrupts most of the bits in Alice's message, but not all of them; asymptotically, about $\frac{1}{e}$ k-tuples will be unaffected, and so there will be a positive residual channel capacity. Given a suitable error correcting code, Alice can still send a message encrypted using Bob's public key. Once Bob has received the message, Alice broadcasts her short one-time key. Bob now applies it to all the messages he has in store; one of them produces a bit string that he can decrypt using his private key. Ward can also now tell that one of the messages he forwarded from Alice to Bob contained suspicious content, namely a random looking string with an error correction code attached, that was most likely an instance of the protocol described here. However, he cannot tell that the message was directed specifically to Bob, as he does not possess Bob's private key.

Chapter 2

Public-Key Encryption

The idea of a public key cryptosystem (PKC) was proposed by Diffie and Hellman in their pioneering paper in 1976. Their revolutionary idea was to enable secure message exchange between sender and receiver without ever having to meet in advance to agree on a common secret key. They proposed the concept of a trapdoor function and how it can be used to achieve a publickey cryptosystem. Shortly there after Rivest, Shamir and Adelman proposed the first candidate trapdoor function, the RSA. The story of modern cryptography followed. The setup for a publickey cryptosystem is of a network of users $u_1 \cdots u_n$ rather than an single pair of users. Each user u in the network has a pair of keys $< P_u, , S_u >$ associated with him, the *public-key* P_u which is published under the users name in a "public directory" accessible for everyone to read, and the privatekey S_u which is known only to u. The pairs of keys are generated by running a keygeneration algorithm. To send a secret message m to u everyone in the network uses the same exact method, which involves looking up P_u, computing $E(P_u, m)$ where E is a public encryption algorithm, and sending the resulting cipher-text c to u. Upon receiving cipher-text c, user u can decrypt by looking up his private key S_u and computing $D(S_u, c)$ where D is a public decryption algorithm. Clearly, for this to work we need that $D(S_u, E(P_u, m)) = m$. A particular PKC is thus defined by a triplet of public algorithms (G, E, D), the key generation, encryption, and decryption algorithms.

2.1 Definition of Public - Key Encryption

Definition 1. A *public key encryption scheme* is a triple, (G,E,D), of probabilistic polynomial time algorithms satisfying the following conditions:

1. *key generation algorithm*: a probabilistic expected polynomial time algorithm G, which, on input 1^k (the security parameter) produces a pair (e, d) where e is called the public key , and d is the corresponding private key. (Notation: $(e, d) \in$

$G(1^k)$). We will also refer to the pair (e, d) a pair of encryption/decryption keys.

2. *An encryption algorithm*: A probabilistic polynomial time algorithm E which takes as input a security parameter 1^k, a publickey e from the range of $G(1^k)$ and string $m \in \{0,1\}^k$ called the message, and produces as output string $c \in \{0,1\}^*$ called the cipher-text. (We use the notation $c \in E(1^k, e, m)$ to denote c being an encryption of message m using key e with security parameter k.

3. *A decryption algorithm*: A probabilistic polynomial time algorithm D that takes as inputs a security parameter 1^k, a privatekey d from the range of $G(1^k)$, and a cipher-text c from the range of $E(1^k, e, m)$, and produces as output a string $m \in \{0,1\}^*$, such that for every pair (e, d) in the range of $G(1^k)$, for every m, for every $c \in D(1^k, e, m)$, the $prob(D(1^k, d, c) \neq m')$ is negligible.

4. This system is "secure".

How to use this definition: To use a publickey encryption scheme (G,E.D) with security parameter 1^k , user A runs $G(1^k)$ to obtain a pair (e, d) of encryption/decryption keys. User A then "publishes" e in a public file, and keeps private d. If anyone wants to send A a message, then need to lookup e and compute $E(1^k, e, m)$. Upon receipt of $c \in E(1^k, e, m)$, A computes message $m = D(1^k, d, c)$.
Note: Messages of length not equal to k (the length of the encryption key) are encrypted by breaking them into blocks of length k and possibly padding the last block. We extend the notation so that

$$E_e(\alpha_1 \cdots \alpha_l \alpha_{l+1}) = E_e(\alpha_1) \cdots E_e(\alpha_l).E_e(\alpha_{l+1}p)$$

where $| \alpha_1 | = \cdots = | \alpha_l | = k, | \alpha_{l+1} | \leq k$, and p is some standard padding of length $k - | \alpha_{l+1} |$ [4].

2.2 Simple Examples of PC: The Trapdoor Function Model

A collection of trapdoor functions, based on oneway functions and trapdoor functions, has been defined as $F = \{f_i : D_i \to D_i\}_{i \in I}$ where $D_i \subseteq \{0,1\}^{|i|}$, and $I$is a set of indices. Recall that $\forall i, f_i$ was easy to compute, but hard to invert; and $\forall i$, there existed t_i such that given t_i and $f_i(x)$, $f_i(x)$ could be inverted in polynomial time. Diffie and Hellman suggested using the supposed existence of trapdoor functions to implement Public Key Cryptosystems as follows:

1. The generator G on security parameter 1^k outputs pairs (f, t_f) where f is a trapdoor function and t_f its associated trapdoor information.

2. For every message $m \in M$, $E(f, m) = f(m)$.

3. Given $c \in E(f, m)$ and t_f , $D(t_f, c) = f^{-1}(c) = f^{-1}(f(m)) = m$.

2.3 Problems with the Trapdoor Function Model

There are several immediate problems which come up in using the trapdoor function model for public key encryption. The main problems which will be:

1. *Special Message Spaces.* The fact that f is a trapdoor function doesn't imply that inverting $f(x)$ when x is special is hard. Namely, suppose that the set of messages that we would like to send is drawn from a highly structured message space such as the English language, or more simply $M = \{0, 1\}$, it may be easy to invert $f(m)$. In fact, it is always easy to distinguish between $f(0)$ and $f(1)$.
2. *Partial Information.* The fact that f is a oneway or trapdoor function doesn't necessarily imply that $f(x)$ hide all information about x. Even a bit of leakage many be too much for some applications. For example, for candidate oneway function $f(p, g, x) = g^x mod\ p$ where p is prime and g is a generator, the least significant bit of x is always easily computable from $f(x)$ [4].
3. *Relationship between Encrypted Messages* Clearly, we may be sending messages which are related to each other in the course of a communication. Some examples are: sending the same secret message to several recipients, or sending the same message (or slight variants) many times. It is thus desirable and sometimes essential that such dependencies remain secret. In the trapdoor function model, it is trivial to see that sending the same message twice is always detectable.

2.4 Problems with Deterministic Encryption in General

The above problems are actually shared by any publickey cryptosystem in which the encryption algorithm is deterministic. It is obvious for problems 1 and 3 above. It is easy to show also for problem 3 as follows. Let E is any deterministic encryption algorithm, we can extract partial information by using something similar to the following predicate:

$$P(x) = \begin{cases} 1 & \text{if E(x) is even} \\ 0 & \text{if E(x) is odd} \end{cases} \tag{2.1}$$

It is clear that we can easily compute this predicate since all we have to do is take the low bit of $E(x)$. Unless $E(x)$ is always even or always odd for all the x's in the message space, we have obtained partial information about x. If $E(x)$ is always even or odd, the low bit of $E(x)$ contains no information. But, some other bit of $E(x)$ must contain some information otherwise the message space is composed of only one message in which case we have total information. Then, simply use that bit instead of the lowest bit and we have a partial information obtaining predicate.

2.5 Probabilistic Public-Key Encryption

In order to build a public key encryption scheme which is polynomial time indistinguishable, we must abandon the trapdoor function PKC model and deterministic algorithms of encryption all together, in favor of probabilistic encryption algorithm. The probabilistic encryption algorithm which we will construct will still assume the existence of trapdoor functions and use them as a primitive building block.
The key to the construction is to first answer a simpler problem: How to securely encrypt single bits. There are two ways to approach this problem. The first is based on trapdoor predicates and the second is based on hard core predicates.

2.5.1 Encrypting Single Bits: Trapdoor Predicates

The Idea: Briefly, a oneway predicate, is a Boolean function which is hard to compute in a very strong sense. Namely, an adversary cannot compute the predicate value better than by taking a random guess. Yet, it is possible to sample the domain of the predicate for elements for which the predicate evaluates to 0 and to 1. A trapdoor predicate possesses the extra feature that there exists some trapdoor information that enables the computation of the predicate. We can construct examples of collection of trapdoor predicates based on the intractability of factoring, RSA inversion and the difficulty of distinguishing quadratic residues from nonresidues.
Definition. Assume that B is a collection of trapdoor predicates. We can now define a public key cryptosystem $(G, E, D)_B$ for sending single bit messages as follows:

1. Key generation algorithm: $G(1^k)$ chooses (i, t_i) (public key is then i and private key is t_i). This is doable by running algorithm S_1.
2. Encryption algorithm: Let $m \in \{0, 1\}$ be the message. Encryption algorithm $E(i, e)$ selects $x \in D_i^m$. (The cipher-text is thus x). This is doable by running algorithm S_2 .
3. Decryption algorithm: $D(c, t_i)$ computes $B_i(c)$. This is doable using A_1 given the trapdoor information.

2.5.2 Encrypting Single Bits: Hard Core Bits

Alternatively, we may take the following perhaps simpler approach, starting directly with trapdoor functions and using their hard core predicates. Recall that a collection of trapdoor permutations is a set $F = \{f_i : D_i \to D_i\}_{i \in I}$ such that:

1. $S_1(1^k)$ samples (i, t_i) where $i \in I$, $| i |= k$ and $| t_i |< p(k)$ for some polynomial

2. $S_2(i)$ samples $x \in D_i$.

3. $\exists$ $PTMA_1$ such that $A_1(i, x) = f_i(x)$.

4. $Pr[A(i, f_i(x)) \in f_i^{-1}(f_i(x))] < \frac{1}{Q(k)} \forall PTMA, \forall Q, \forall k > k_0$

5. $\exists$ $PTMA_2$ such that $A_2(i, t_i, f_i(x)) = x, \forall x \in D_i, i \in I$.

Definition: Given a collection F with hard core predicates B, define public key cryptosystem $(G, E, D)_B$ for sending a single bit as follows:

- Key generation algorithm: $G(1^k)$ chooses pair $< i, t_i >$ by running $S_1(1^k)$
- Encryption algorithm: $E(i, m)$ chooses at random an $x \in D_i$ such that $B_i(x) = m$, and output as a cipher-text $f_i(x)$. Using the Goldreich Levin construction of a hard core predicate, simply choose x, r such that the inner product of x and r is m and output $f(x) \circ r$.
- Decryption algorithm: To decrypt $c = f_i(x)$, given i and t_i, the decryption algorithm $D(t_i, c)$ compute $B_i(f_i^{-1}(c)) = B_i(x) = m$. Using the Goldreich Levin construction this amounts to given $c = f_i(x) \circ r$ to computing the inner product of x and r.

Chapter 3

Provably Secure Steganography

3.1 Basics

Definition 1. A function $f : N \to (0,1)$ is said to be *negligible* if for everyc > 0, for all sufficiently large n, $f(n) < 1/n^c$.

Note 1. The concatenation of string s1 and string s2 will be denoted by $s1 \parallel s2$, and "Parses as $s_1 \parallel s_2 \parallel \cdots \parallel s_l$" to mean, separate s into strings $s_1 \parallel s_2 \cdots s_l$
Note 2. Let $U(k)$ denote the uniform distribution on k bit strings, and $U(L,l)$ denote the uniform distribution on functions from L bit strings to l bit strings.
Note 3. If X is finite a set, then let $U(X)$ denote the uniform distribution on X.

3.1.1 Notations from Cryptography

Let $F : \{0,1\}^k \times \{0,1\}^L \to \{0,1\}^l$ denote a family of functions. Let A be an oracle which acts probabilistic adversary. Define the *prf-advantage* of A over F as:

$$\mathbf{Adv}_F^{prf}(A) = \left\| \Pr_{K \leftarrow U(k), r \leftarrow \{0,1\}^*}[A_r^{F_K(\cdot)} = 1] - \Pr_{g \leftarrow U(L,l), r \leftarrow \{0,1\}^*}[A_r^g = 1] \right\|$$

where r is the string of random bits used by adversary A. Define the insecurity of F as:

$$\mathbf{InSec}_F^{prf}(t,q) = \max_{A \in A(t,q)} \left\{ \mathbf{Adv}_F^{prf}(A) \right\}$$

where $A(t,q)$ denotes the set of adversaries taking at most t steps steps and making at most q oracle queries. Then F is a *(t,q,ϵ)-pseudo-random function* if $\mathbf{InSec}_F^{prf}(t,q) \leq \epsilon$. Suppose that $l(k)$ and $L(k)$ are polynomials.
A sequence $\{F_k\}_{k \in N}$ of families $F_k : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{l(k)}$ is called *pseudo-random* if for all polynomially bounded adversaries A, $\mathbf{Adv}_{F_k}^{prf}(A)$ is negligible in k.
Let $E : K \times R \times P \to C$ be a probabilistic private key encryption scheme, which

maps a random number and an $\mid m \mid$-bit plain-text to a cipher-text.
Consider a game in which an adversary A is given access to an oracle which is either:

1. E_K for $K \leftarrow U(K)$; that is, an oracle which given a message m, uniformly selects random bits R and returns $E_K(R{:}\, m)$; or
2. $g(\cdot) = U(\mid E_K(\cdot) \mid)$; that is, an oracle which on any query ignores its input and returns a uniformly selected output of the appropriate length.

Let $A(t, q, l)$ be the set of adversaries A which make q queries to the oracle of at most l bits and run for t time steps. Define the CPA advantage of A against E as:

$$\mathbf{Adv}_E^{cpa}(A) = \mid \| \Pr_{K \leftarrow U(k), s, r \leftarrow \{0,1\}^*} [A_r^{E_{K,s}} = 1] - \Pr_{g \leftarrow, r \leftarrow \{0,1\}^*} [A_r^g = 1] \| \mid$$

where $E_{K,s}$ denotes E_K with random bit source s. Define the insecurity of E as:

$$\mathbf{InSec}_E^{cpa}(t, q, l) = \max_{A \in A(t,q,l)} \left\{ \mathbf{Adv}_E^{cpa}(A) \right\}$$

Then(t, q, l, ϵ) - *indistinguishable from random bits under chosen plain-text attack* if $\mathbf{InSec}_E^{cpa}(t, q, l) \leq \epsilon$.
Definition 2. A sequence of cryptosystems $\{E_k\}_{k \in N}$ is called indistinguishable from random bits under chosen plain-text attack (INDS-CPA) if for every PPTM A, $\mathbf{Adv}_{E_k}^{cpa}(A)$ is *negligible* in k.
Let C be a distribution with finite support X.
Definition 3. The minimum entropy of C,$H_\infty(C)$,is

$$H_\infty(C) = \min_{x \in X} \left\{ log_2 \frac{1}{Pr_C[x]} \right\}$$

3.1.2 Steganography

Steganography will be thought of as a game between the warden, Ward, and the inmate, Alice. The goal of Alice is to pass a secret message to Bob over a communication channel (known to Ward). The goal of Ward is to detect whether a secret message is being passed.
Definition 4. A *channel* is a distribution on bit sequences where each bit is also timestamped with monotonically non-decreasing time value. Formally, a channel is a distribution with support $(\{0, 1\}, t1), (\{0, 1\}, t2), \cdots$, where $\forall i > 0 : t_{i+1} \geq t_i$.The definition of a channel is sufficiently general to encompass nearly any form of communication. Anyone communicating on a channel can be regarded as implicitly drawing from the channel, so we can assume the existence of an oracle capable of drawing from the channel. In fact, we will assume something stronger: 'an oracle that can

partially draw from the channel a (finite, fixed length) sequence of bits. This oracle can draw from the channel in steps and at any point the draw is conditioned on what has been drawn so far. Let C_h be the channel distribution conditional on the history h of already drawn timestamped bits. And let C_h^b be the marginal channel distribution over the next block of b timestamped bits conditional on the history h. Intuitively, C_h^b is a distribution on the next b time stamped bits conditioned on the history h. Fix b. Assume the existence of an oracle which can draw from C_h^b . And call such a partial draw a "block". We will require that the channel satisfy a minimum entropy constraint for all blocks:

$$\forall h \; drawn \; from \; C : H_\infty(C_h^b) > 1$$

This partial draw will be conditional on all past draws and so we can regard a sequence of partial draws as a draw from the channel.

Definition 5. (Stegosystem) A steganographic protocol, or stegosystem, is a pair of probabilistic algorithms $S = (SE, SD)$:

1. SE takes a key $K \in \{0,1\}^k$, a string $m \in \{0,1\}^*$(the hidden-text), a message history h, and an oracle $M(h)$ which samples blocks according to a channel distribution C_h^b.
2. $SE^M(K, m, h)$ returns a sequence of blocks $c_1 \parallel c_2 \parallel \cdots \parallel c_l$(the stegotext) each of which is an element from the support of the channel C_h^{l*b}being sampled by the oracle M.
3. SD takes a key K, a sequence of blocks $c_1 \parallel c_2 \cdots \parallel c_l$,a message history h, and an oracle $M(h)$
4. $SD_M(K, c, h)$ returns a hiddentext m.
5. There must be a polynomial $p(k) > k$ such that SE^M and SD^M also satisfy the relationship:

$$\forall m, \mid m \mid < p(k) : Pr(SD^M(K, SE^M(K, m, h), h) = m) \geq \frac{2}{3}$$

where the randomization is over any coin tosses of SE^M, SD^M, and M.
Note: The probability of failure for the stegosystem of 1/3 is choosen in order to include a wide range of possible stegosystems. In general, given a protocol with any reasonable probability of failure, we can boost the system to a very low probability of failure using error-correcting codes.
Although all of our oracle-based protocols will work with the oracle. $M(h)$ Rejection Sampling function. $RS^{M,F} : \{0,1\}^* \times N \to \{0,1\}$.

Procedure $RS^{M,F}$:

- **Input:** target x, iteration *count*
- i = 0
- repeat: $c \leftarrow M$,increment i
- until F(c) = x or i = count
- **Output: c**

The function RS simply samples from the distribution provided by the sample oracle M until $F(M) = x$. The function will return c satisfying $F(c) = x$ or the count-th sample from M. Note that an iteration count is used to bound the worst case running time of RS and that RS may fail to return a c satisfying $F(c) = x$.

Comment. We have taken the approach of assuming a channel which can be drawn from freely by the stegosystem; most current proposals for stegosystems act on a single sample from the channel (one exception is). While it may be possible to define a stegosystem which is steganographically secret or robust and works in this style, this is equivalent to a system in this model which merely makes a single draw on the channel distribution. Further, And believe is that the lack of reference to the channel distribution may be one of the reasons for the failure of many such proposals in the literature.

In practice, this oracle is also the weakest point of all these constructions. And assume the existence of a perfect oracle: one that can perform independent draws, one that can be rewound, etc. This assumption can be justified in some cases, but not in others. If the oracle is a human, the human may not be able to perform independent draws from the channel as is required by the function RS. A real world Warden would use this to his advantage.
remark: The protocols are shown to be secure under the assumption that the oracle is perfect. Decoding algorithm, SD, is defined to have access to the oracle $M(h)$. This is a general definition, and there are cases in which this access will not be necessary. Protocols in which SD needs no access to $M(h)$ are clearly preferred[5].

3.1.3 Steganographic Secrecy

A passive warden, W, is an adversary which plays the following game:

1. W is given access to an oracle $M(h)$ which samples blocks (one at a time) from the distribution C_h^b , for past histories h drawn from the channel. W makes as many draws from $M(h)$ as it likes.

2. W is given access to a second oracle which is either $SE(K,\cdot,\cdot)$ or $O(m,h)$ defined by $O(m,h) \leftarrow C_h^{|SE(K,m,h)|}$. Ward W makes at most q queries totaling l bits (of hiddentext) to this oracle.

3. W outputs a bit.

We define W's advantage against a stegosystem S by

$$\mathbf{Adv}_{S,C}^{SS}(W) = \mid \Pr_{K,M,r,SM}[W^{M,SE(K,\cdot,\cdot)} = 1] - \Pr_{r,M,O}[W^{M,O(\cdot,\cdot)_r} = 1] \mid$$

where the warden uses random bits r. Define the insecurity of S by

$$\mathbf{InSec}_{S,C}^{SS}(t,q,l) = \max_{W \in W(t,q,l)} \left\{ \mathbf{Adv}_{S,C}^{SS}(W) \right\}$$

where $W(t,q,l)$ denotes the set of all adversaries which make at most q queries totaling at most l bits (of hiddentext) and running in time at most t.
Definition 6. (Steganographic secrecy) A Stegosystem $S = (SE, SD)$ is called (t,q,l,ϵ) *steganographically secret against chosen hiddentext attack* for the channel C $((t,q,l,\epsilon) - SS - CHA - C)$ if

$$\mathbf{InSec}_{S,C}^{SS}(t,q,l) \leq \epsilon$$

Definition 7.(Universal Steganographic Secrecy) A Stegosystem S is called (t,q,l,ϵ)-*universally steganographically secret against chosen hiddentext attack* $((t,q,l,\epsilon) - USS - CHA)$ if it is $(t,q,l,\epsilon) - SS - CHA - C$ for every channel distribution C that satisfies $H_\infty(C_h^b) > 1$ for all h drawn from C.
Definition 8. A sequence of stegosystems $\{S_k\}_k \in N$ is called universally steganographically secret if for every channel distribution C and for every PPTM W, $\mathbf{Adv}_{S(k),C}^{SS}(W)$ is negligible in k.

Note. Steganographic secrecy can be thought of roughly as encryption which is indistinguishable from arbitrary distributions C.

3.2 Stateless Steganographic Secrecy

The following protocol described here can be found in[5] satisfies above definition for steganographic secrecy. This protocol (up to small differences) is not new and can be found in [1]; an information theoretic version of the protocol can also be found in [2].

Definition 9. A function $f : D \to R$ is said to be unbiased function on a distribution C if:

$$\forall r \in R, h, Pr_{d \leftarrow C_h^b}[f(d) = r] = \frac{1}{|R|}$$

Let $f : (\{0,1\}, t)^b \to \{0,1\}$ be a public function which is unbiased on C.

Note. The assumption that there exists an f satisfying this definition does not hold for all channels.

Let $E_K(\cdot,\cdot)$ and $D_K(\cdot)$ denote the encryption and decryption functions for a cryptosystem which is indistinguishable from random bits under chosen plaintext attack (i.e., IND\$-CPA) [16]. Suppose Alice and Bob share a key $K \in \{0,1\}^k$. The following procedures allow encoding and decoding of messages in a manner which is steganographically secret under chosen hiddentext attack for the channel distribution C.

Construction 1.(Stateless Steganographic Secrecy)

```
Procedure: S1.Encode:
Input: key K, hiddentext m, history h
Let $s = E_K(m)$
Parse s as $s_1^1 \| \cdots \| s_l^1$
for $i = 1 \ldots l$ do
    $c_i \leftarrow RS^{M(h), f(\cdot)}(s_i, | K |)$
    set $h = h \| c_i$

Output: $c_1 \| c_2 \| \cdots \| c_l$
```

```
Procedure: S1.Decode:
Input: key K, Stegotext $c$, history $h$
Parse $c$ as $c_1^b \| \cdots \| c_l^b$
for $i = 1 \ldots l$ do
    Set$s_i \leftarrow= f(c_i)$
set$s = s_1 \| s_2 \| \cdots \| s_l$.
Output:$D(K, s)$
```

Note that the execution time for SD is essentially linear in l and the execution time for SE is an expected $O(l)$. Also note that the assumption of an unbiased function implies that an error correcting code as in construction 1 is unnecessary: the probability of failure here is $l/2^{|K|}$. Clearly, the assumption of an unbiased f is quite strong. In practice, perfect unbiasedness may be hard to meet and stronger assumptions may be required. For example, when f is a pseudorandom function, under the assumption of a large minimum entropy in the channel. f will be nearly unbiased.

Lemma 1 *For any warden $W \in W(t, q, l)$, we can construct a corresponding adversary A where*

$$\boldsymbol{Adv}^{SS}_{S2,C}(W) = \boldsymbol{Adv}^{cpa}_{E}(A)$$

The running time of A is at most $t + O(kl)$, and A makes q encryption queries for a total of l bits of plaintext.

Proof. Suppose we are given a program W for distinguishing between the output of construction 1 and the ordinary traffic of the channel, and an oracle for sampling blocks from that channel. We can construct a program A which plays the CPA game – distinguishing an $E_K(R, \cdot)$ oracle from a uniform g oracle – with the same advantage as W. A simply runs the encoding procedure S1.Encode using its oracle in place of $E_K(R, \cdot)$. Consider the following two cases:

1. $O(m) = E_K(m)$. Then the stegotexts output by the encoding procedure will be identically distributed to stegotexts resulting from the normal use of construction 1.

2. $O(m) = g(m)$ is chosen uniformly from strings of appropriate length. Then the stegotexts output by the encoding procedure will be identically distributed to a history-dependent sample from the channel distribution C^b_h . This follows by the fact that f is unbiased on C and the parsed substrings $g(m)_i$ are uniformly distributed on {0,1}.

3.3 Stateful Steganographic Secrecy

In this a stateful protocol which is universally steganographically secret is present. This protocol[5] also is not new and can be found (up to small differences) in [1], for instance. The novel aspect is the proof of steganographic secrecy within this model.

Assume Alice and Bob share a channel and let C denote the channel distribution. Let $M(h)$ be an oracle which draws blocks (one at a time) from the conditional distribution C^b_h . Let $F_K(\cdot, \cdot)$ denote a pseudorandom function family indexed by $k = | K |$ key bits which maps a pair consisting of a d-bit number and b timestamped bits to the set {0,1}, i.e., $F : \{0,1\}^k \times \{0,1\}^d \times (\{0,1\}, t)^b \leftarrow \{0,1\}$. Let Alice and

Bob share a secret key $K \in \{0,1\}^k$ and a synchronized d bit counter N. Let Enc and Dec be the encoding and decoding procedures, respectively, for an error-correcting code which corrects up to a $3/8 + \mathbf{InSec}_F^{prf}(O(l), 2l)$ fraction of errors, and which has stretch function $L(\cdot)$. The following procedures allow Alice and Bob to encode and decode a hiddentext and to send it via their shared channel.

Construction 2.(Stateful Steganographic Secrecy)

```
Procedure: S2.Encode:
Input: key K, hiddentext $m'$, history h
Let $m = E_K(m')$
Parse $m$ as $m_1^1 \| \cdots \| m_l^1$
for $i = 1 \ldots l$ do
$c_i \leftarrow RS^{M(h),F_K(N,\cdot)}(m_i, 2)$
set $h = h \| c_i$
increment N
Output: $c_1 \| c_2 \| \cdots \| c_l$
```

```
Procedure: S2.Decode:
Input: key K, Stegotext $c$,
Parse $c$ as $c_1^b \| \cdots \| c_l^b$
for $i = 1 \ldots l$ do
Set $m_i \leftarrow= F_K(N, c_i)$
increment N
let $m = m_1 \| m_2 \| \cdots \| m_l$.
Output: $Dec(m)$
```

The idea behind this construction is simple. First, the encoding algorithm splits $m = Enc(m')$ into single bits. Then it draws blocks from the channel at most twice, trying to find a block c_1 such that $F_K(N, c_1) = m_1$. After that it tries to find c_2 such that $F_K(N+1, c_2) = m_2$, and c_3, and so on. This continues until all bits of m have been encoded. The decoding algorithm simply applies $F_K(N+i-1, \cdot)$ to all the c_i's.

Note that the encoding procedure actually has some probability of failure per bit. In particular, RS might not return a c_i such that $F_K(N+i-1, \cdot) = m_i$, because RS "gives up" after 2 timesteps. The probability of failure dependent on the sampling oracle M and the evaluation oracle F.

Lemma 1 *the probability of failure of RS in the above procedure is bounded above by* $3/8 + \mathbf{InSec}_{F_k}^{prf}(O(l), 2l)$.

Proof. Assume the channel has symbols $\{S_1, \cdots, S_k\}$ with probabilities $p_1, \cdots, p_k$, and assume F is a truly random function. Consider the following experiment:
Draw from the channel to get a symbol S. If $F(S, N) = 0$ output S. Otherwise draw again from the channel and output the result.
Denote the outcome of this experiment by S_E. Let D be the event that after drawing twice from the channel you get a different symbol (a non-collision), and let $\overline{D}$ denote the event that after drawing twice from the channel you get the same symbol. Then:

$$Pr[f(S_E, N) = 0] = \frac{1}{2} + \frac{1}{4}Pr[D]$$

This is because the right side is the sum of the probabilities of the following disjoint events:

- The first draw results in a symbol that maps to zero.
- The first draw results in a symbol that maps to 1 and the second draw results in a different symbol that maps to 0.

Let S_i be a symbol with highest probability $(p_i \geq p_j forallj)$, then

$$Pr[\overline{D}] = p_1^2 + \cdots + p_k^2 \leq p_i(p_1 + \cdots + p_k)$$

So $Pr[D] \geq (1 - p_i)$, and we have that

$$Pr[f(S_E) = 0] \geq \frac{1}{2} + \frac{1}{4}(1 - p_i)$$

$$\geq \frac{1}{2} + \frac{1}{8}$$

In practice F is not random function, but rather pseudorandom function so the probability of failure of RS is bounded by $3/8 + \mathbf{InSec}^{prf}_{F_K}(O(l), 2l)$.

Theorem 1 *Let $k = \mid K \mid$. For any $l \leq 2^d$:*

$$\mathbf{InSec}^{SS}_{S1,C}(t, q, l) \leq \mathbf{InSec}^{prf}_{F}(t + O(L(l)), L(l))$$

Proof. For any warden, W, running in time t and making q queries totaling l bits, we construct a corresponding PRF adversary A, where

$$\mathbf{Adv}^{SS}_{S1,C}(W) = \mathbf{Adv}^{prf}_{F}(A)$$

The running time of A is the running time of warden W plus the time of rejection sampling (RS) : $O(L(l))$ in the worst case. The number of calls to the sampling oracle, $M(h)$, is at most $2L(l)$.

A_f simply runs W, emulating the encoding procedure S2.Encode using the function oracle f in place of $F_K(\cdot,\cdot)$. Note that when f is a uniformly chosen random function, the output of $RS^{M(h),f}(\cdot,2)$ is distributed identically to the channel distribution C_h^2. Similarly, when f is chosen from $F_K(\cdot,\cdot)$, the output of $RS^{M(h),f}(\cdot,2)$ is distributed identically to the output of Construction 1, by the definition of the construction. So the advantage is:

$$\begin{aligned}\mathbf{Adv}_F^{prf}(A) &= \left\| \Pr_{K\leftarrow U(k),r\leftarrow\{0,1\}^*}[A_r^{F_K(\cdot,\cdot)}=1] - \Pr_{g\leftarrow U(L,l),r\leftarrow\{0,1\}^*}[A_r^g=1]\right\| \\ &= \left\| \Pr_{K,M,r,SM}[W^{M,SE(K,\cdot,\cdot)}=1] - \Pr_{r,M,O}[W^{M,O(\cdot,\cdot)_r}=1]\right\| \\ &= \mathbf{Adv}_{S1,C}^{SS}(W)\end{aligned}$$

And the theorem follows by the definition of insecurity.

Corollary 1 *If $F_K(\cdot,\cdot)$ is pseudorandom then construction 2 is universally steganographically secret against chosen-hiddentext attacks.*

Proof. $F_K(\cdot,\cdot)$ pseudorandom implies that for all polynomially bounded adversaries A, $\mathbf{Adv}_{F_k}^{prf}(A)$is negligible in k. The definition of insecurity and theorem 1 implies that for any cover channel, C, the advantage of a warden will be negligible in k.This, in turn, implies the corollary.

Chapter 4

Provably secure Stego-Algorithm

How should we encrypt arbitrary length messages[4]? The first answer is to simply encrypt each bit individually using one of the above methods. Before considering whether this is wise from an efficiency point of view, we need to argue that it indeed will produce a encryption scheme which is polynomial time indistinguishable. This requires reflection, as even through every bit individually is secure, it can be the case that some predicate computed on all the bits is easily computable, such as the exclusive of the bits. This turns out luckily not to be the case.
Definition : Define a probabilistic encryption based on trapdoor collection F with hard core bit B to be $PE = (G, E, D)$ where:

1. $G(1^k)$ chooses (i, t_i) by running $S_1(1^k)$ (Public key is i, private key is t_i).
2. Let $m = m_1 \cdots m_k$ where $m_j \in \{0, 1\}$ be the message.
   $E(i, m)$ encrypts m as follows:
   Choose $x_j \in D_i$ such that $B_i(x_j) = m_j$ for $j = 1, \cdots, k$.
   Output $c = f_i(x_1) \cdots f_i(x_k)$.
3. Let $c = y_1, \cdots, y_k$ where $y_i \in D_i$ be the cipher-text.
   $D(t_i, c)$ decrypts c as follows:
   Compute $m_j = B_i(f_i^{-1}(y_j))$ for $j = 1, \cdots, k$.
   Output $m = m_1 \cdots m_k$.

4.1 Efficient Probabilistic Encryption

How efficient are the probabilistic schemes? In some schemes the cipher-text may be longer than the clear-text by a factor proportional to the security parameter. However, it has been shown using later ideas on pseudorandom number generation how to start with trapdoor functions and build a probabilistic encryption scheme that is polynomial **time** secure for which the cipher-text is longer than the clear-text by only an additive factor. The most efficient probabilistic encryption scheme is due to Blum and Goldwasser and is comparable with the RSA deterministic encryption scheme in speed and data expansion. Recall, that private key encryption seemed to be much more efficient. Indeed, in practice the public key methods are often used to transmit a secret session key between two participants which have never met, and subsequently the secret session key is used in conjunction with a privatekey encryption method. We first describe a probabilistic public key cryptosystem based on any trapdoor function collection which suffers only from a small additive bandwidth expansion.

As in the previous probabilistic encryption PE, we begin with a collection of trapdoor permutations $F = \{f_i : D_i \rightarrow D_i\}$ with hard core predicates $B = \{B_i : D_i \rightarrow \{0, 1\}\}$. We consider that $D_i \subseteq \{0,1\}^k$, where $k = \mid i \mid$.

Then $EPE = (G, E, D)$ is our PKC based on F with:

Key Generation: $G(1^k) = S_1(1^k) = (i, t_i)$. The public key is i, and the secret key is t_i .

Encryption Algorithm: To encrypt m, $E(i, m)$ runs as follows, where $l = \mid m \mid$:

1. Choose $r \in D_i$ at random.
2. Compute $f_i(r), f_i^2(r), \cdots, f_i^l(r)$.
3. Let $p = B_i(r)B_i(f_i(r))B_i(f_i^2(r)) \cdots B_i(f_i^{l-1}(r))$.
4. Output the cipher-text $c = (p \oplus m, f_i^l(r))$.

Decryption Algorithm: To decrypt a cipher-text $c = (m', a)$, $D(t_i, c)$ runs as follows, where $l = \mid m' \mid$:

1. Compute r such that $f_i^l(r) = a$. We can do this since we can invert f_i using the trapdoor information, t_i , and this r is unique since f_i is a permutation.
2. Compute the pad as above for encryption: $p = B_i(r)B_i(f_i(r)) \cdots B_i(f_i^{l-1}(r))$.
3. Output decrypted message $m = m' \oplus p$.

4.2 Provably secure Stego-Algorithm.

An implementation of EPE with cost equal to the cost of RSA In this section.

we consider a particular implementation of EPE as efficient as RSA. This uses for F a subset of Rabin's trapdoor functions Recall that we can reduce Rabin's functions to permutations if we only consider the Blum primes, and restrict the domain to the set of quadratic residues. In fact, we will restrict our attention to primes of the form

$p \equiv 7 \; mod \; 8$.
Let $N = \{n \mid n = pq : \mid p \mid = \mid q \mid, p, q \equiv 7 \; mod \; 8\}$. Then let $F = \{f_n : D_n \to D_n\}_{n \in N}$, where $f_n(x) \equiv x^2 \; mod \; n$, and $D_n = Q_n = \{y \mid y \equiv x^2 \; mod \; n\}$. Because $p, q \equiv 3 \; mod \; 4$, we have that f_n is a permutation on D_n. $B_n(x)$ is the least significant bit (LSB) of x, which is a hard core bit if and only if factoring is difficult, i.e., the Factoring Assumption is true. Then consider the EPE (G, E, D), with:
Key Generation: $G(1^k) = (n, (p, q))$ where $pq = n \in N$, and $\mid n \mid = k$. Thus n is the public key, and (p, q) is the secret key.
Encryption: $E(n, m')$, where $l = \mid m' \mid$

1. Choose $r \in Q_n$ randomly.
2. Compute $r^2, r^4, r^8, \cdots r^{2^l} (mod \; n)$.
3. Let $p = LSB(r)LSB(r^2)LSB(r^4) \cdots LSB(r^{2^{l-1}})$.
4. Output $s = (m' \oplus p, r^{2^l} mod \; n)$.

The cost of encryption is $O(k^2.l)$.
Decryption: $D((p, q), s)$, where $s = (m'', a)$, $l = \mid m'' \mid$

1. Compute r such that $r^{2^l} \equiv mod \; n$.
2. Compute $p = LSB(r)LSB(r^2)LSB(r^4) \cdots LSB(r^{2^{l-1}})$.
3. Output $m' = m'' \oplus p$.

The proposed Stego - Algorithm.

```
Procedure: SE.Encode:
Input: key K, hiddentext $m'$, history h
Let $s = E_K(m')$
Parse $s$ as $s_1^1 \parallel \cdots \parallel s_l^1$
for $i = 1 \ldots l$ do
$c_i \leftarrow RS^{M(h), F_K(\cdot)}(s_i, \mid k \mid)$
set $h = h \parallel c_i$
Output: $c_1 \parallel c_2 \parallel \cdots \parallel c_l$
Procedure: SD.Decode:
Input: key K, Stegotext c,
Parse $c$ as $c_1^b \parallel \cdots \parallel c_l^b$
for $i = 1 \ldots l$ do
Set $s_i \leftarrow F_K(c_i)$
let $s = s_1 \parallel s_2 \parallel \cdots \parallel s_l$.
Output: $Dec(s)$.
```

Note: Here we can find the value of r with the help of trapdoor t_i in Decryptoin as follows[4]:
Since $p \equiv 7 \bmod 8$, we have $p = 8t + 7$ and $q = 8s + 7$ and we can find $r_p \equiv \sqrt[2^l]{a} \equiv a^{(2t+2)^l} \bmod p$, similarly $r_q \equiv \sqrt[2^l]{a} \equiv a^{(2s+2)^l} \bmod q$, and using *Chinese Remainder Th eorem*[12] we can find r.

The assumption of an unbiased f is quite strong. In general, perfect unbiased may be hard to meet for all channels and stronger assumptions required [8]. For example, when f is a pseudorandom function, under the assumptions of large minimum entropy in the channel, f will be nearly unbiased.

Chapter 5

Conclusion

Here we analyzed the work of Hopper, Langford, Von Ahn who gave the theoretical concept of provably secure steganography recently in CRYPTO-2002. Here we proposed a Provably Stego-Algorithm which is based on Rabin's trapdoor functions with restrict the domain to the set of quadratic residues. And can be implemented efficiently. More over the cost of proposed Stego-Algorithm is equal to the cost of RSA algorithm.

Bibliography

[1] Ross Anderson. Stretching the limits of steganography. *Lecture Notes In Computer Science, Proceedings of the First International Workshop on Information Hiding,* pages 39–48, 1996.

[2] Christian Cachin. An information-theoretic model for steganography. *International Association for Cryptologic Research,* 028, 2002.

[3] Kahn David. *The Code Breakers.* Macmillan, 1967.

[4] Shafi Goldwasser and Bellare. *Lecture Notes in Cryptography.* MIT Laboratory of Computer Science, 2001.

[5] Nicholas J. Hopper, Jhon Langford, and Luis von Ahn. Provably secure steganography. *In Proceedings of CRYPTO-2002,* 2002.

[6] Stefan Katzenbeisser and Fabien A. P. Petitcolas. *Information Hiding Techniques for Steganography and digital Watermarking.* Artech House Books, 1999.

[7] NF Maxemchuk. Electronic document distribution. *AT & T Technical Journal,* 1998.

[8] Leonoid Reyzin and Scott Russell. More efficient provably secure steganography. *International Association for Cryptologic Research,* 2003/093:093, 2003.

[9] Anderson Ross J. and Petitcolas Fabien A. On the limits of steganography. *IEEE Journel of Selected Areas in Communications,* 16(4), 1998.

[10] B Schneier. *Applied Cryptography.* Wiley, 1995.

[11] G. J. Simmons. The prisoner's problem and subliminal channel. *In Procedings of CRYPTO'83,* 1984.

[12] Douglas R. Stinson. *Cryptography: Theory and Practice.* CRC Press, 2002.

[13] RG van Schyndel and AZ Tirkel CF Osborne. A digital watermark. *International conference on Imageprocessing, (IEEE),* 1994.