Toward a theory of Steganography

Nicholas J. Hopper

CMU-CS-04-157

July 2004

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Manuel Blum, Chair
Avrim Blum
Michael Reiter
Steven Rudich
David Wagner, U.C. Berkeley

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This material is based upon work partially supported by the National Science Foundation under Grants CCR-0122581 and CCR-0058982 (The Aladdin Center) and an NSF Graduate Fellowship; the Army Research Office (ARO) and the Cylab center at Carnegie Mellon University; and a Siebel Scholarship.

Keywords: Steganography, Cryptography, Provable Security

Abstract

Informally, *steganography* refers to the practice of hiding secret messages in communications over a public channel so that an eavesdropper (who listens to all communications) cannot even tell that a secret message is being sent. In contrast to the active literature proposing new concrete steganographic protocols and analysing flaws in existing protocols, there has been very little work on formalizing steganographic notions of security, and none giving complete, rigorous proofs of security in a satisfying model.

My thesis initiates the study of steganography from a cryptographic point of view. We give a precise model of a communication channel and a rigorous definition of steganographic security, and prove that relative to a channel oracle, secure steganography exists if and only if one-way functions exist. We give tightly matching upper and lower bounds on the maximum *rate* of any secure stegosystem. We introduce the concept of steganographic key exchange and public-key steganography, and show that provably secure protocols for these objectives exist under a variety of standard number-theoretic assumptions. We consider several notions of *active attacks* against steganography, show how to achieve each under standard assumptions, and consider the relationships between these notions. Finally, we extend the concept of steganograpy as covert communication to include the more general concept of covert *computation*.

Acknowledgments

I profusely thank Manuel Blum for five years of constant support, interesting discussions, and strange questions. I hope I am able to live up to his standard of advising.

Much of this work was done in collaboration with Luis von Ahn and John Langford. The work was “born” on our car trip back to Pittsburgh from CCS 2001 in Philadelphia. I owe many thanks to both for their challenging questions and simplifying explanations.

My other committee members - Avrim Blum, Steven Rudich, Michael Reiter, and David Wagner - all made valuable comments about my thesis proposal and earlier versions of this thesis; I'm sure that it is stronger because of them.

And of course, I am extremely thankful to my wife Jennie for many things, not the least of which was following me to Pittsburgh; and my daughter Allie for being herself.

For Jennie and Allie.

Contents

Chapter 1

Introduction

This dissertation focuses on the problem of steganography: how can two communicating entities send secret messages over a public channel so that a third party cannot detect the presence of the secret messages? Notice how the goal of steganography is different from classical encryption, which seeks to conceal the *content* of secret messages: steganography is about hiding the very existence of the secret messages.

Steganographic "protocols" have a long and intriguing history that goes back to antiquity. There are stories of secret messages written in invisible ink or hidden in love letters (the first character of each sentence can be used to spell a secret, for instance). More recently, steganography was used by prisoners, spies and soldiers during World War II because mail was carefully inspected by both the Allied and Axis governments at the time [38]. Postal censors crossed out anything that looked like sensitive information (e.g. long strings of digits), and they prosecuted individuals whose mail seemed suspicious. In many cases, censors even randomly deleted innocent-looking sentences or entire paragraphs in order to prevent secret messages from being delivered. More recently there has been a great deal of interest in digital steganography, that is, in hiding secret messages in communications between computers.

The recent interest in digital steganography is fueled by the increased amount of communication which is mediated by computers and by the numerous potential commercial applications: hidden information could potentially be used to detect or limit the unauthorized propagation of the innocent-looking "carrier" data. Because of this, there have been numerous proposals for protocols to hide data in channels containing pictures [37, 40], video [40, 43, 61], audio [32, 49], and even typeset text [12]. Many of these protocols are extremely clever and rely heavily on domain-specific properties of these channels. On the other hand, the literature on steganography also contains many clever attacks which detect the use of such protocols. In addition, there is no clear consensus in the literature about what it should mean for a stegosystem to be secure; this ambiguity makes it unclear whether it is even possible to have a secure protocol for steganography.

The main goal of this thesis is to rigorously investigate the open question: "under what conditions do secure protocols for steganography exist?" We will give rigorous cryptographic definitions of steganographic security in multiple settings against several different types of adversary, and we will demonstrate necessary and sufficient conditions for security in each setting, by exhibiting protocols which are secure under these conditions.

1.1 Cryptography and Provable Security

The rigorous study of *provably secure* cryptography was initiated by Shannon [58], who introduced an information-theoretic definition of security: a cryptosystem is secure if an adversary who sees the *ciphertext* - the scrambled message sent by a cryptosystem - receives no additional information about the *plaintext* - the unscrambled content. Unfortunately, Shannon also proved that any cryptosystem which is perfectly secure requires that if a sender wishes to transmit N bits of plaintext data, the sender and the receiver must share at least N bits of random, secret data - the *key*. This limitation means that only parties who already possess secure channels (for the exchange of secret keys) can have secure communications.

To address these limitations, researchers introduced a theory of security against *computationally limited* adversaries: a cryptosystem is computationally secure if an adversary who sees the ciphertext cannot compute (in, e.g. polynomial time) any additional information about the plaintext than he could without the ciphertext [31]. Potentially, a cryptosystem which could be proven secure in this way would allow two parties who initially share a very small number of secret bits (in the case of public-key cryptography, zero) to subsequently transmit an essentially unbounded number of message bits securely.

Proving that a system is secure in the computational sense has unfortunately proved to be an enormous challenge: doing so would resolve, in the negative, the open question of whether $P = NP$. Thus the cryptographic theory community has borrowed a tool from complexity theory: reductions. To prove a cryptosystem secure, one starts with a computational problem which is presumed to be intractable, and a model of how an adversary may attack a cryptosystem, and proves via reduction that computing any additional information from a ciphertext is equivalent to solving the computational problem. Since the computational problem is assumed to be intractable, a computationally limited adversary capable of breaking the cryptosystem would be a contradiction and thus should not exist. In general, computationally secure cryptosystems have been shown to exist if and only if "one-way functions," which are easy to compute but computationally hard to invert, exist. Furthermore, it has been shown that the difficulty of a wide number of well-investigated number-theoretic problems would imply the existence of one-way functions, for example the problem of computing the factors of a product of two large primes [13], or computing discrete logarithms in a finite field [14].

Subsequent to these breakthrough ideas [13, 31], cryptographers have investigated a wide variety of different ways in which an adversary may attack a cryptosystem. For example, he may be allowed to make up a plaintext message and ask to see its corresponding ciphertext, (called a chosen-plaintext attack), or even to make up a ciphertext and ask to see what the corresponding plaintext is (called a chosen-ciphertext attack [48, 52]). Or the adversary may have a different goal entirely [8, 23, 39] - for example, to modify a ciphertext so that if it previously said "Attack" it now reads as "Retreat" and vice-versa. We will draw on this practice to consider the security of a steganographic protocol under several different kinds of attack.

These notions will be explored in further detail in Chapter 2.

1.2 Previous work on theory of steganography

The scientific study of steganography in the open literature began in 1983 when Simmons [59] stated the problem in terms of communication in a prison. In his formulation, two inmates, Alice and Bob, are trying to hatch an escape plan. The only way they can communicate with each other is through a public channel, which is carefully monitored by the warden of the prison, Ward. If Ward detects any encrypted messages or codes, he will throw both Alice and Bob into solitary confinement. The problem of steganography is, then: how can Alice and Bob cook up an escape plan by communicating over the public channel in such a way that Ward doesn't suspect anything "unusual" is going on.

Anderson and Petitcolas [6] posed many of the open problems resolved in this thesis. In particular, they pointed out that it was unclear how to prove the security of a steganographic protocol, and gave an example which is similar to the protocol we present in Chapter 3. They also asked whether it would be possible to have steganography without a secret key, which we address in Chapter 4. Finally, they point out that while it is easy to give a loose upper bound on the rate at which hidden bits can be embedded in innocent objects, there was no known lower bound.

Since the paper of Anderson and Petitcolas, several works [16, 44, 57, 66] have addressed information-theoretic definitions of steganography. Cachin's work [16, 17] formulates the problem as that of designing an encoding function so that the relative entropy between *stegotexts*, which encode hidden information, and independent, identically distributed samples from some innocent-looking *covertext* probability distribution, is small. He gives a construction similar to one we describe in Chapter 3 but concludes that it is computationally intractable; and another construction which is provably secure but relies critically on the assumption that all orderings of covertexts are equally likely. Cachin also points out several flaws in other published information-theoretic formulations of steganography.

All information-theoretic formulations of steganography are severely limited, however, because it is easy to show that information-theoretically secure steganography implies information-theoretically secure encryption; thus any secure stegosystem with

N bits of secret key can encode at most N hidden bits. In addition, techniques such as public-key steganography and robust steganography are information-theoretically impossible.

1.3 Contributions of the thesis

The primary contribution of this thesis is a rigorous, cryptographic theory of steganography. The results which establish this theory fall under several categories: symmetric-key steganography, public-key steganography, steganography with active adversaries, steganographic rate, and steganographic *computation*. Here we summarize the results in each category.

Symmetric Key Steganography.

A symmetric key stegosystem allows two parties with a shared secret to send hidden messages undetectably over a public channel. We give cryptographic definitions for symmetric-key stegosystems and steganographic secrecy against a passive adversary in terms of indistinguishability from a probabilistic *channel* process. By giving a construction which provably satisfies these definitions, we show that the existence of a one-way function is sufficient for the existence of secure steganography relative to any channel. We also show that this condition is necessary by demonstrating a construction of a one-way function from any secure stegosystem.

Public-Key Steganography

Informally, a public-key steganography protocol allows two parties, who have never met or exchanged a secret, to send hidden messages over a public channel so that an adversary cannot even detect that these hidden messages are being sent. Unlike previous settings in which provable security has been applied to steganography, public-key steganography is information-theoretically *impossible*. We introduce computational security conditions for public-key steganography similar to those for the symmetric-key setting, and give the first protocols for public-key steganography and steganographic key exchange that are provably secure under standard cryptographic assumptions.

Steganography with active adversaries

We consider the security of a stegosystem against an adversary who actively attempts to subvert its operation by introducing new messages to the communication between Alice and Bob. We consider two classes of such adversaries: *disrupting* adversaries and *distinguishing* adversaries. Disrupting adversaries attempt to prevent Alice and Bob from communicating steganographically, subject to some set of publicly-known restrictions; we give a formal definition of *robustness* against such an attack and give the first construction of a provably robust stegosystem. Distinguishing adversaries introduce additional traffic between Alice and Bob in hopes of tricking them into revealing their use of steganography; we consider the security of symmetric- and public-key stegosystems against active distinguishers and give constructions which are secure against such adversaries. We also show that *no stegosystem can be simultaneously secure against both disrupting and distinguishing active adversaries.*

Bounds on steganographic rate

The *rate* of a stegosystem is defined by the (expected) ratio of hiddentext size to stegotext size. Prior to this work there was no known lower bound on the achievable rate (since there were no provably secure stegosystems), and only a trivial upper bound. We give an upper-bound MAX in terms of the number of samples from a probabilistic channel oracle and the minimum-entropy of the channel, and show that this upper bound is tight by giving a provably secure symmetric-key stegosystem with rate $(1 - o(1))$MAX. We also give an upper bound RMAX on the rate achievable by a robust stegosystem and exhibit a construction of a robust stegosystem with rate $(1 - \epsilon)$RMAX for any $\epsilon > 0$.

Covert Computation

We introduce the novel concept of *covert two-party computation.* Whereas ordinary secure two-party computation only guarantees that no more knowledge is leaked about the inputs of the individual parties than the result of the computation, covert two-party computation employs steganography to yield the following additional guarantees: (A) no outside eavesdropper can determine whether the two parties are performing the computation or simply communicating as they normally do; (B) before learning $f(x_A, x_B)$, neither party can tell whether the other is running the protocol; (C) after the protocol concludes, each party can only determine if the other ran the protocol insofar as they can distinguish $f(x_A, x_B)$ from uniformly chosen random bits. Covert two-party computation thus allows the construction of protocols that return $f(x_A, x_B)$ only when it equals a certain value of interest (such as "Yes, we are romantically interested in each other") but for which *neither party can determine whether the other even ran the protocol whenever $f(x_A, x_B)$ does not equal the value of interest.* We introduce security definitions for covert two-party computation and we construct protocols with provable security based on the Decisional Diffie-Hellman assumption.

A steganographic design methodology

At a higher level, the technical contributions of this thesis suggest a powerful design methodology for steganographic security goals. This methodology stems from the observation that the uniform channel is universal for steganography: we give a transformation from an arbitrary protocol which produces messages indistinguishable from uniform random bits (given an adversary's view) into a protocol which produces messages indistinguishable from an arbitrary channel distribution (given the adversary's view). Thus, in order to hide information from an adversary in a given channel, it is sufficient to design a protocol which hides the information among pseudorandom bits and apply our transformation. Examples of this methodology appear in Chapters 3, 4, 5, and 7; and the explicit transformation for a general task along with a proof of its security is given in chapter 7, Theorem 7.5.

1.4 Roadmap of the thesis

Chapter 2 establishes the results and notation we will use from cryptography, and describes our model of innocent communication. Chapter 3 discusses our results on symmetric-key steganography and relies heavily on the material in Chapter 2. Chapter 4 discusses our results on public-key steganography, and can be read independently of chapter 3. Chapter 5 considers active attacks against stegosystems; section 5.1 depends on material in Chapters 2 and 3, while the remaining sections also require some familiarity with the material in Chapter 4. Chapter 6 discusses the rate of a stegosystem, and depends on materials in Chapter 3, while the final section also requires material from section 5.1. Finally, in Chapter 7 we extend steganography from the concept of hidden communication to hidden computation. Chapter 7 depends only on the material in chapter 2. Finally, in Chapter 8 we suggest directions for future research.

Chapter 2

Model and Definitions

In this chapter we will introduce the notation and concepts from cryptography and information theory that our results will use. The reader interested in a more general treatment of the relationships between the various notions presented here is referred to the works of Goldreich [25] and Goldwasser and Bellare [30].

2.1 Notation

We will model all parties by Probabilistic Turing Machines (PTMs). A PTM is a standard Turing machine with an additional read-only "randomness" tape that is initially set so that every cell is a uniformly, independently chosen bit. If A is a PTM, we will denote by $x \leftarrow A(y)$ the event that x is drawn from the probability distribution defined by A's output on input y for a uniformly chosen random tape. We will write $A_r(y)$ to denote the output of A with random tape fixed to r on input y.

We will often make use of Oracle PTMs (OPTM). An OPTM is a PTM with two additional tapes: a "query" tape and a "response" tape; and two corresponding states $Q_{\text{query}}, Q_{\text{response}}$. An OPTM runs with respect to some oracle O, and when it enters state Q_{query} with value y on its query tape, it goes in one step to state Q_{response}, with $x \leftarrow O(y)$ written to its "response" tape. If O is a probabilistic oracle, then $A^O(y)$ is a probability distribution on outputs taken over both the random tape of A and the probability distribution on O's responses.

We denote the length of a string or sequence s by $|s|$. We denote the empty string or sequence by ε. The concatenation of string s_1 and string s_2 will be denoted by $s_1\|s_2$, and when we write "Parse s as $s_1\|_{t_1} s_2\|_{t_2} \cdots \|_{t_{l-1}} s_l$" we mean to separate s into strings $s_1, \ldots s_l$ where each $|s_i| = t_i$ and $s = s_1\|s_2\| \cdots \|s_l$. We will assume the use of efficient and unambiguous pairing and unpairing operators on strings, so that (s_1, s_2) may be uniquely interpreted as the pairing of s_1 with s_2, and is not the same as $s_1\|s_2$. One example of such an operation is to encode (s_1, s_2) by a prefix-free encoding of $|s_1|$, followed by s_1, followed by a prefix-free encoding of $|s_2|$ and then s_2. Unpairing then reads $|s_1|$, reads that many bits from the input into s_1, and repeats the process for s_2.

We will let U_k denote the uniform distribution on $\{0,1\}^k$. If X is a finite set, we will denote by $x \leftarrow X$ the action of uniformly choosing x from X. We denote by $U(L, l)$ the uniform distribution on functions $f : \{0,1\}^L \to \{0,1\}^l$. For a probability distribution D, we denote the support of D by $[D]$. For an integer n, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

2.2 Cryptography and Provable Security

Modern cryptography makes use of *reductions* to prove the security of protocols; that is, to show that a protocol P is secure, we show how an attacker violating the security of P can be used to solve a problem Q which is believed to be intractable. Since solving Q is believed to be intractable, it then follows that violating the security of P is also intractable. In this section, we will give examples from the theory of symmetric cryptography to illustrate this approach, and introduce the notation to be used in the rest of the dissertation.

2.2.1 Computational Indistinguishability

Let $\mathcal{X} = \{X_k\}_{k\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_k\}_{k\in\mathbb{N}}$ denote two sequences of probability distributions such that $[X_k] = [Y_k]$ for all k. Many cryptographic questions address the issue of distinguishing between samples from $\mathcal{X}$ and samples from $\mathcal{Y}$. For example, the distribution $\mathcal{X}$ could denote the possible encryptions of the message "Attack at Dawn" while $\mathcal{Y}$ denotes the possible encryptions of "Retreat at Dawn;" a cryptanalyst would like to distinguish between these distributions as accurately as possible, while a cryptographer would like to show that they are hard to tell apart. To address this concept, cryptographers have developed several notions of indistinguishability. The simplest is the statistical distance:

Definition 2.1. (Statistical Distance) Define the statistical distance between $\mathcal{X}$ and $\mathcal{Y}$ by

$$\Delta_k(\mathcal{X}, \mathcal{Y}) = \frac{1}{2} \sum_{x \in [X_k]} |\Pr[X_k = x] - \Pr[Y_k = x]| \ .$$

If $\Delta(X, Y)$ is small, it will be difficult to distinguish between X and Y, because most outcomes occur with similar probability under both distributions.

On the other hand, it could be the case that $\Delta(X, Y)$ is large but X and Y are still difficult to distinguish by some methods. For example, if X_k is the distribution on k-bit even-parity strings starting with 0 and Y_k is the distribution on k-bit even-parity strings starting with 1, then an algorithm which attempts to distinguish X and Y based on the parity of its input will fail, even though $\Delta(X, Y) = 1$. To address this situation, we define the *advantage* of a program:

Definition 2.2. (Advantage) We will denote the *advantage* of a program A in distinguishing $\mathcal{X}$ and $\mathcal{Y}$ by

$$\mathbf{Adv}_A^{\mathcal{X},\mathcal{Y}}(k) = |\ \Pr[A(X_k) = 1] - \Pr[A(Y_k) = 1]\ | \ .$$

Thus in the previous example, for any program A that considers only $\sum_i s_i \bmod 2$, it will be the case that $\mathbf{Adv}_A^{\mathcal{X},\mathcal{Y}}(k) = 0$.

While the class of adversaries who consider only the parity of a string is not very interesting, we may consider more interesting classes: for example, the class of all adversaries with running time bounded by $t(k)$.

Definition 2.3. (Insecurity) We denote the *insecurity* of X, Y by

$$\mathbf{InSec}_{\mathcal{X},\mathcal{Y}}(t, k) = \max_{A \in TIME(t(k))} \left\{ \mathbf{Adv}_A^{\mathcal{X},\mathcal{Y}}(k) \right\}$$

and we say that X_k and Y_k are (t, ϵ) indistinguishable if $\mathbf{InSec}_{\mathcal{X},\mathcal{Y}}(t, k) \leq \epsilon$.

If we are interested in the case that $t(k)$ is bounded by some polynomial in k, then we say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, written $\mathcal{X} \approx \mathcal{Y}$, if for every $A \in TIME(poly(k))$, there is a negligible function ν such that $\mathbf{Adv}_A^{\mathcal{X},\mathcal{Y}}(k) \leq \nu(k)$. (A function $\nu : \mathbb{N} \to (0, 1)$ is said to be *negligible* if for every $c > 0$, for all sufficiently large n, $\nu(n) < 1/n^c$.)

We will make use, several times, of the following (well-known) facts about statistical and computational distance:

Proposition 2.4. Let $\Delta(X, Y) = \epsilon$. Then for any probabilistic program A,

$$\Delta(A(X), A(Y)) \leq \epsilon \ .$$

Proof.

$$\begin{aligned}
\Delta(A(X), A(Y)) &= \frac{1}{2} \sum_x |\Pr[A(X) = x] - \Pr[A(Y) = x]| \\
&= \frac{1}{2} \sum_x \left| 2^{-|r|} \sum_r (\Pr[A_r(X) = x] - \Pr[A_r(Y) = x]) \right| \\
&\leq \frac{1}{2} 2^{-|r|} \sum_r \sum_x |\Pr[A_r(X) = x] - \Pr[A_r(Y) = x]| \\
&\leq \frac{1}{2} \max_r \sum_x |\Pr[A_r(X) = x] - \Pr[A_r(Y) = x]| \\
&\leq \frac{1}{2} \max_r \sum_x \sum_{y \in A_r^{-1}(x)} |\Pr[X = y] - \Pr[Y = y]| \\
&\leq \Delta(X, Y) \ .
\end{aligned}$$

□

Proposition 2.5. For any t, $\mathbf{InSec}_{X,Y}(t, k) \leq \Delta(X, Y)$

Proof. Let $A \in TIME(t)$ be any program with range $\{0, 1\}$. Then we have that

$$\begin{aligned}
\mathbf{Adv}_A^{X,Y}(k) &= |\Pr[A(X) = 1] - \Pr[A(Y) = 1]| \\
&= |(1 - \Pr[A(X) = 0]) - (1 - \Pr[A(Y) = 0])| \\
&= |\Pr[A(X) = 0] - \Pr[A(Y) = 0]| \\
&= \frac{1}{2}(|\Pr[A(X) = 0] - \Pr[A(Y) = 0]| + |\Pr[A(X) = 1] - \Pr[A(Y) = 1]|) \\
&= \Delta(A(X), A(Y)) \ .
\end{aligned}$$

And thus, by the previous proposition, $\mathbf{Adv}_A^{X,Y}(k) \leq \Delta(X, Y)$. Since this holds for every A, we then have that

$$\mathbf{InSec}_{X,Y}(t, k) = \max_{A \in TIME(t)} \left\{ \mathbf{Adv}_A^{X,Y}(k) \right\} \leq \Delta(X, Y) \ .$$

□

Proposition 2.6. For any $m \in \mathbb{N}$, $\mathbf{InSec}_{X^m,Y^m}(t, k) \leq m\mathbf{InSec}_{X,Y}(t + (m-1)T, k)$, where $T = \max\{\text{Time to sample from X}, \text{Time to sample from Y}\}$.

Proof. The proof uses a "hybrid" argument. Consider any $A \in TIME(t)$; we wish to bound $\mathbf{Adv}_A^{X^m,Y^m}(k)$. To do so, we define a sequence of hybrid distributions $Z_0, \ldots, Z_m$, where $Z_0 = X^m$, $Z_m = Y^m$, and $Z_i = (Y^i, X^{m-i})$. We will consider the "experiment" of using A to distinguish Z_i from Z_{i+1}.

Notice that starting from the definition of advantage, we have:

$$\begin{aligned}
\mathbf{Adv}_A^{X^m,Y^m}(k) &= |\Pr[A(X^m) = 1] - \Pr[A(Y^m) = 1]| \\
&= |\Pr[A(Z_0) = 1] - \Pr[A(Z_m) = 1]| \\
&= |(\Pr[A(Z_0) = 1] - \Pr[A(Z_1) = 1]) + (\Pr[A(Z_1) = 1] - \Pr[A(Z_2) = 1]) \\
&\quad + \cdots + (\Pr[A(Z_m) = 1] - \Pr[A(Z_{m-1}) = 1])| \\
&\leq \sum_{i=1}^{m} |\Pr[A(Z_i) = 1] - \Pr[A(Z_{i-1}) = 1]| \\
&= \sum_{i=1}^{m} \mathbf{Adv}_A^{Z_{i-1},Z_i}(k)
\end{aligned}$$

Now notice that for each i, there is a program B_i which distinguishes X from Y with the same advantage as A has in distinguishing Z_{i-1} from Z_i: on input S, B_i draws

$i-1$ samples from Y, $m-i$ samples from X, and runs A with input (Y^{i-1}, S, X^{m-i}). If $S \leftarrow X$, then $\Pr[B_i(S)=1] = \Pr[A(Z_{i-1})=1]$, because the first $i-1$ samples in A's input will be from Y, and the remaining samples will be from X. On the other hand, if $S \leftarrow Y$, then $\Pr[B_i(S)=1] = \Pr[A(Z_i)=1]$, because the first i samples in A's input will be from Y. So we have:

$$\begin{aligned}\mathbf{Adv}_{B_i}^{X,Y}(k) &= |\Pr[B_i(X)=1] - \Pr[B_i(Y)=1]| \\ &= |\Pr[A(Z_{i-1})=1] - \Pr[A(Z_i)=1]| \\ &= \mathbf{Adv}_A^{Z_{i-1},Z_i}(k)\ .\end{aligned}$$

And therefore we can bound A's advantage in distinguishing X^m,Y^m by

$$\mathbf{Adv}_A^{X^m,Y^m}(k) \leq \sum_{i=1}^{m} \mathbf{Adv}_{B_i}^{X,Y}(k)\ .$$

Now since B_i takes as long as A to run (plus time at most $(m-1)T$ to draw the additional samples from X,Y), it follows that

$$\mathbf{Adv}_{B_i}^{X,Y}(k) \leq \mathbf{InSec}_{X,Y}(t+(m-1)T, k)\ ,$$

so we can conclude that

$$\mathbf{Adv}_A^{X^m,Y^m}(k) \leq m\mathbf{InSec}_{X,Y}(t+(m-1)T, k)\ .$$

Since the theorem holds for any $A \in TIME(t)$, we have that

$$\mathbf{InSec}_{X^m,Y^m}(t,k) \leq \max_{A \in TIME(t)} \left\{ \mathbf{Adv}_A^{X^m,Y^m}(k) \right\} \leq m\mathbf{InSec}_{X,Y}(t+(m-1)T, k)\ ,$$

as claimed. □

The style of proof we have used for this proposition, in which we attempt to state as tightly as possible the relationship between the "security" of two related problems without reference to asymptotic analysis, is referred to in the literature as concrete security analysis. In this dissertation, we will give concrete security results except in Chapter 8, in which the concrete analysis would be too cumbersome.

2.2.2 Universal Hash Functions

A Universal Hash Family is a family of functions $H : \{0,1\}^l \times \{0,1\}^m \to \{0,1\}^n$ where $m \geq n$, such that for any $x_1 \neq x_2 \in \{0,1\}^m$ and $y_1, y_2 \in \{0,1\}^n$,

$$\Pr_{Z \leftarrow U_l}[H(Z, x_1) = y_1 \wedge H(Z, x_2) = y_2] = 2^{-2n} \ .$$

Universal hash functions are easy to construct for any m, n with $l = 2m$, by considering functions of the form $h_{a,b}(x) = ax + b$, over the field $GF(2^m)$, with truncation to the least significant n bits. It is easy to see that such a family is universal, because truncation is regular, and the full-rank system $ax_1 + b = y_1$, $ax_2 + b = y_2$ has exactly one solution over $GF(2^m)$, which is selected with probability 2^{-2m}. We will make use of universal hash functions to convert distributions with large minimum entropy into distributions which are indistinguishable from uniform.

Definition 2.7. (Entropy) Let $\mathcal{D}$ be a distribution with finite support X. Define the *minimum entropy* of $\mathcal{D}$, $H_\infty(\mathcal{D})$, as

$$H_\infty(\mathcal{D}) = \min_{x \in X} \left\{ \log_2 \frac{1}{\Pr_{\mathcal{D}}[x]} \right\} \ .$$

Define the *Shannon entropy* of $\mathcal{D}$, $H_S(\mathcal{D})$ by

$$H_S(\mathcal{D}) = \mathop{E}_{x \leftarrow \mathcal{D}} \left[-\log_2 \Pr_{\mathcal{D}}[x] \right] \ .$$

Lemma 2.8. (Leftover Hash Lemma, [33]) Let $H : \{0,1\}^l \times \{0,1\}^m \to \{0,1\}^n$ be a universal hash family, and let $X : \{0,1\}^m$ satisfy $H_\infty(X) \geq k$. Then

$$\Delta((Z, H(Z, X)), (Z, U_n)) \leq 2^{-(k-n)/2+1}$$

.

As a convention, we will sometimes refer to H as a family of functions and identify elements of H by their index, e.g., when we say $h \in H$, then $h(x)$ refers to $H(h, x)$.

2.2.3 Pseudorandom Generators

Let $G = \left\{ G_k : \{0,1\}^k \to \{0,1\}^{l(k)} \right\}_{k \in \mathbb{N}}$ denote a sequence of functions, with $l(k) > k$. Then G is a pseudorandom generator (PRG) if $G(U_k) \approx U_{l(k)}$. More formally, define the PRG-advantage of A against G by:

$$\mathbf{Adv}^{\mathsf{prg}}_{A,G}(k) = \left|\Pr[A(G(U_k)) = 1] - \Pr[A(U_{l(k)}) = 1]\right|$$

And the PRG-Insecurity of G by

$$\mathbf{InSec}^{\mathsf{prg}}_{G}(t,k) = \max_{A \in TIME(t(k))} \left\{\mathbf{Adv}^{\mathsf{prg}}_{A,G}(k)\right\} .$$

Then G_k is a (t,ϵ)-secure PRG if $\mathbf{InSec}^{\mathsf{prg}}_{G}(t,k) \leq \epsilon$, and G is a PRG if for every $A \in TIME(poly(k))$, there is a negligible μ such that $\mathbf{Adv}^{\mathsf{prg}}_{A,G}(k) \leq \mu(k)$.

Pseudorandom generators can be seen as the basic primitive on which symmetric cryptography is built. If G is a (t,ϵ)-PRG, then $G(U_k)$ can be used in place of $U_{l(k)}$ for any application, and the loss in security against $TIME(t)$ adversaries will be at most ϵ. It was shown by Håstad *et al* [33] that asymptotically, PRGs exist if and only if one-way functions (OWFs) exist; thus when we say that the existence of a primitive is equivalent to the existence of one-way functions, we may show it by giving reductions to and from PRGs.

2.2.4 Pseudorandom Functions

Let $F : \{0,1\}^k \times \{0,1\}^L \to \{0,1\}^l$ denote a family of functions. Informally, F is a pseudorandom function family (PRF) if F and $U(L,l)$ are indistinguishable by oracle queries. Formally, let A be an oracle probabilistic adversary. Define the *prf-advantage of A over F* as

$$\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) = \left| \Pr_{K \leftarrow U(k)}[A^{F_K(\cdot)}(1^k) = 1] - \Pr_{f \leftarrow U(L,l)}[A^{f}(1^k) = 1] \right| .$$

Define the insecurity of F as

$$\mathbf{InSec}^{\mathsf{prf}}_{F}(t,q,k) = \max_{A \in \mathcal{A}(t,q)} \left\{\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k)\right\}$$

where $\mathcal{A}(t,q)$ denotes the set of adversaries taking at most t steps and making at most q oracle queries. Then F_k is a (t,q,ϵ)-*pseudorandom function* if $\mathbf{InSec}^{\mathsf{prf}}_{F}(t,q,k) \leq \epsilon$. Suppose that $l(k)$ and $L(k)$ are polynomials. A sequence $\{F_k\}_{k \in \mathbb{N}}$ of families $F_k : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{l(k)}$ is called *pseudorandom* if for all polynomially bounded adversaries A, $\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k)$ is negligible in k. We will sometimes write $F_k(K,\cdot)$ as $F_K(\cdot)$.

We will make use of the following results relating PRFs and PRGs.

Proposition 2.9. Let $F_k : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{l(k)}$ be a PRF. Let $q = \lceil \frac{k+1}{l(k)} \rceil$. Define $G_k : \{0,1\}^k \to \{0,1\}^{k+1}$ by $G(X) = F_X(0) \| F_X(1) \| \cdots \| F_X(q-1)$. Then

$$\mathbf{InSec}_G^{\mathsf{prg}}(t,k) \le \mathbf{InSec}_F^{\mathsf{prf}}(t+q,q,k)$$

Proof. Consider an arbitrary PRG adversary A. We will construct a PRF adversary B with the same advantage against F as A has against G. B has oracle access to a function f. B makes q queries to f, constructing the string $s = f(0) \| \cdots \| f(q-1)$, and then returns the output of A on s. If f is a uniformly chosen function, the string s is uniformly chosen; thus

$$\Pr[B^f(1^k) = 1] = \Pr[A(U_{k+1}) = 1] \ .$$

If f is an element of F, then the string s is chosen exactly from $G(U_k)$. In this case, we have

$$\Pr[B^{F_K}(1^k) = 1] = \Pr[A(G(U_k)) = 1] \ .$$

Combining the cases gives us

$$\begin{aligned} \mathbf{Adv}_{B,F}^{\mathsf{prf}}(k) &= \left| \Pr[B^{F_K}(1^k) = 1] - \Pr[B^f(1^k) = 1] \right| \\ &= \left| \Pr[A(G(U_k)) = 1] - \Pr[A(U_{k+1}) = 1] \right| \\ &= \mathbf{Adv}_{A,G}^{\mathsf{prg}}(k) \end{aligned}$$

Since B runs in the same time as A plus the time to make q oracle queries, we have by definition of insecurity that

$$\mathbf{Adv}_{B,F}^{\mathsf{prf}}(k) \le \mathbf{InSec}_F^{\mathsf{prf}}(t+q,q,k) \ ,$$

and thus, for every A, we have

$$\mathbf{Adv}_{A,G}^{\mathsf{prg}}(k) \le \mathbf{InSec}_F^{\mathsf{prf}}(t+q,q,k) \ ,$$

which yields the stated theorem. $\square$

Intuitively, this proposition states that a pseudorandom function can be used to construct a pseudorandom generator. This is because if we believe that F is pseudorandom, we must believe that $\mathbf{InSec}_F^{\mathsf{prf}}(t,q,k)$ is small, and therefore that the insecurity of the construction G, $\mathbf{InSec}_G^{\mathsf{prg}}(k)$ is also small.

Proposition 2.10. ([27], Theorem 3) There exists a function family $\mathcal{F}^G : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^k$ such that

$$\mathbf{InSec}^{\mathsf{prf}}_{\mathcal{F}^G}(t,q,k) \leq qk\mathbf{InSec}^{\mathsf{prg}}_{G}(t + qkTIME(G), k) \ .$$

2.2.5 Encryption

A symmetric cryptosystem $\mathcal{E}$ consists of three (randomized) algorithms:

- $\mathcal{E}.\mathtt{Generate} : 1^k \to \{0,1\}^k$ generates shared *keys* $\in \{0,1\}^k$. We will abbreviate $\mathcal{E}.\mathtt{Generate}(1^k)$ by $G(1^k)$, when it is clear which encryption scheme is meant.
- $\mathcal{E}.\mathtt{Encrypt} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ uses a key to transform a *plaintext* into a *ciphertext*. We will abbreviate $\mathcal{E}.\mathtt{Encrypt}(K, \cdot)$ by $E_K(\cdot)$.
- $\mathcal{E}.\mathtt{Decrypt} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ uses a key to transform a ciphertext into the corresponding plaintext. We will abbreviate $\mathcal{E}.\mathtt{Decrypt}(K, \cdot)$ by $D_K(\cdot)$.

Such that for all keys K, $\mathcal{E}.\mathtt{Decrypt}(K, \mathcal{E}.\mathtt{Encrypt}(K, m)) = m$. Informally, we will say that a cryptosystem is secure if, after viewing encryptions of plaintexts of its choosing, an adversary cannot distinguish ciphertexts from uniform random strings. This is slightly different from the more standard notion in which it is assumed that encryptions of distinct plaintexts are indistinguishable.

To formally define the security condition for a cryptosystem, consider a game in which an adversary A is given access to an oracle $\mathcal{O}$ which is either:

- E_K for $K \leftarrow G(1^k)$; that is, an oracle which given a message m, returns a sample from $E_K(m)$; or
- $\$(\cdot)$; that is, an oracle which on query m ignores its input and returns a uniformly selected string of length $|E_K(m)|$.

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

$$\mathbf{Adv}^{\mathsf{cpa}}_{A,\mathcal{E}}(k) = \left|\Pr[A^{E_K}(1^k) = 1] - \Pr[A^{\$}(1^k) = 1]\right|$$

where the probabilities are taken over the oracle draws and the randomness of A. Define the insecurity of E as

$$\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, q, l, k) = \max_{A \in \mathcal{A}(t,q,l)} \left\{ \mathbf{Adv}^{\mathsf{cpa}}_{A,\mathcal{E}}(k) \right\} .$$

Then $\mathcal{E}$ is (t, q, l, k, ϵ)*-indistinguishable from random bits under chosen plaintext attack* if $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, q, l, k) \leq \epsilon$. $\mathcal{E}$ is called (computationally) *indistinguishable from random bits under chosen plaintext attack* (IND\$-CPA) if for every PPTM A, $\mathbf{Adv}^{\mathsf{cpa}}_{A,\mathcal{E}}(k)$ is negligible in k.

It was shown by [33] that the existence of secure symmetric cryptosystems is equivalent to the existence of OWFs.

Proposition 2.11. ([36], Theorem 4.3) Let $\mathcal{E}$ be a symmetric cryptosystem. Then there is a generator $G^{\mathcal{E}}$ such that G is a PRG if $\mathcal{E}$ is IND\$-CPA.

Proposition 2.12. Let $F : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ be a function family. Define the cryptosystem $\mathcal{E}^F$ as follows:

- $G(1^k) \leftarrow U_k$.
- $E_K(m_1 \cdots m_l) = c \leftarrow U_k \| F_K(c+1) \oplus m_1 \| \cdots \| F_K(c+l) \oplus m_l$.
- $D_K(c \| x_1 \cdots x_l) = F_K(c+1) \oplus x_1 \| \cdots \| F_K(c+l) \oplus x_l$.

Then

$$\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}^F}(t, q, l, k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + 2l, l, k) + \frac{ql}{2^{k-1}} .$$

Proof. Let A be a chosen-plaintext attacker for $\mathcal{E}$. We will construct a PRF attacker for F which has advantage at least

$$\mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) \geq \mathbf{Adv}^{\mathsf{cpa}}_{A,\mathcal{E}}(k) - \frac{ql}{2^{k-1}} .$$

B will run in time $t + 2l$ and make l queries to its function oracle, so that

$$\mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + 2l, l, k) ,$$

which will yield the result.

B's strategy is to play the part of the encryption oracle in A's chosen-plaintext attack game. Thus, B will run A, and whenever A makes an encryption query, B will produce a response using its function oracle, which it will pass back to A. At the conclusion of the chosen-plaintext game, A produces an output bit, which B will use for its output. It remains to describe how B will respond to A's encryption queries. B will do so by executing the encryption program E_K from above, but using its function oracle in place of F_K. Thus, on a query $m_1 \cdots m_l$, B^f will choose a $c \leftarrow U_k$, and give A the response $c \| f(c+1) \oplus m_1 \| \cdots \| f(c+l) \oplus m_l$.

Let us bound the advantage of B. In case B's oracle is chosen from F_K, B will perfectly simulate an encryption oracle to A. Thus

$$\Pr[B^{F_K}(1^k) = 1] = \Pr[A^{E_K}(1^k) = 1] \ .$$

Now suppose that B's oracle is a uniformly chosen function, and let NC denote the event that B does not query its oracle more than once on any input, and let C denote the complement of NC - that is, the event that B queries its oracle at least twice on at least one input. Conditioned on NC, every bit that B returns to A is uniformly chosen, for a uniform choice of f, subject to the condition that none of the leading values overlap, an event we will denote by $\mathsf{N\$}$, and which has identical probability to NC. In this case B perfectly simulates a random-bit oracle to A, giving us

$$\Pr[B^f(1^k) | \mathsf{NC}] = \Pr[A^{\$}(1^k) = 1 | \mathsf{N\$}] \ .$$

By conditioning on NC and C, we find that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) &= \Pr[B^{F_K}(1^k) = 1] - \Pr[B^f(1^k) = 1] \\
&= \Pr[A^{E_K}(1^k) = 1] - \big(\Pr[B^f(1^k) = 1 | \mathsf{NC}] \Pr[\mathsf{NC}] \\
&\qquad\qquad + \Pr[B^f(1^k) = 1 | \mathsf{C}] \Pr[\mathsf{C}]\big) \\
&\geq \Pr[A^{E_k}(1^k) = 1] - \Pr[A^{\$}(1^k) = 1 \wedge \mathsf{N\$}] - \Pr[\mathsf{C}] \\
&\geq \Pr[A^{E_k}(1^k) = 1] - \Pr[A^{\$}(1^k) = 1] - \Pr[\mathsf{C}] \\
&= \mathbf{Adv}^{\mathsf{cpa}}_{A,\mathcal{E}}(k) - \Pr[\mathsf{C}] \ ,
\end{aligned}$$

where we assume without loss of generality that $\Pr[A^{E_K}(1^k) = 1] \geq \Pr[A^{\$}(1^k) = 1]$. To finish the proof, we need only to bound $\Pr[\mathsf{C}]$.

To bound the probability of the event C, let us further subdivide this event. During the attack game, A will make q queries that B must answer, so that B chooses q k-bit values $c_1, \ldots, c_q$ to encrypt messages of length $l_1, \ldots, l_q$; Let us denote by NC_i the event that after the i^{th} encryption query made by A, B has not made any duplicate queries to its function oracle f; and let C_i denote the complement of NC_i. We will show that

$$\Pr[\mathsf{C}_i | \mathsf{NC}_{i-1}] \leq \frac{il_i + \sum_{j<i} l_j}{2^k} ,$$

and therefore we will have

$$\begin{aligned}
\Pr[\mathsf{C}] &= \Pr[\mathsf{C_q}] \\
&\leq \Pr[\mathsf{C_q} | \mathsf{NC}_{q-1}] + \Pr[\mathsf{C}_{q-1}] \\
&\leq \sum_{i=1}^{q} \Pr[\mathsf{C}_i | \mathsf{NC}_{i-1}] \\
&\leq \frac{1}{2^k} \sum_{i=1}^{q} \left(il_i + \sum_{j<i} l_j \right) \\
&\leq \frac{1}{2^k} \left(\sum_{i=1}^{q} il_i + ql \right) \\
&\leq \frac{1}{2^k} \left(q \sum_{i=1}^{q} l_i + ql \right) \\
&= \frac{2ql}{2^k}
\end{aligned}$$

Which establishes the desired bound, given the bound on $\Pr[\mathsf{C}_i | \mathsf{NC}_{i-1}]$. To establish this conditional bound, fix any choice of the values $c_1, \ldots, c_{i-1}$. The value c_i will cause a duplicate input to f if there is some c_j such that $c_j - l_i \leq c_i \leq c_j + l_j$, which happens with probability $(l_i + l_j)/2^k$, since c_i is chosen uniformly. Thus by the union bound, we have that

$$\Pr[\mathsf{C_i} | \mathsf{NC}_{i-1}] \leq 2^{-k} \sum_{j<i} (l_i + l_j)$$

and rearranging gives the stated bound:

$$\Pr[\mathsf{C_i} | \mathsf{NC_{i-1}}] \leq 2^{-k} (il_i + \sum_{j<i} l_j) .$$

$\square$

2.3 Modeling Communication - Channels

We seek to define steganography in terms of indistinguishability from a "usual" or innocent-looking pattern of communication. In order to do so, we must characterize this pattern. We begin by supposing that Alice and Bob communicate via *documents:*

Definition 2.13. (Documents) Let D be an efficiently recognizable, prefix-free set of strings, or *documents*.

As an example, if Alice and Bob are communicating over a computer network, they might run the TCP protocol, in which case they communicate by sending "packets" according to a format which specifies fields like a source and destination address, packet length, and sequence number.

Once we have specified what kinds of strings Alice and Bob send to each other, we also need to specify the probability that Ward will assign to each document. The simplest notion might be to model the innocent communications between Alice and Bob by a *stationary* distribution: each time Alice communicates with Bob, she makes an independent draw from a probability distribution $\mathcal{C}$ and sends it to Bob. Notice that in this model, all orderings of the messages output by Alice are equally likely. This does not match well with our intuition about real-world communications; if we continue the TCP analogy, we notice, for example, that in an ordered list of packets sent from Alice to Bob, each packet should have a sequence number which is one greater than the previous; Ward would become very suspicious if Alice sent all of the odd-numbered packets first, and then all of the even.

Thus, we will use a notion of a channel which models a prior distribution on the entire sequence of communication from one party to another:

Definition 2.14. A *channel* is a distribution on sequences $s \in D^{\Omega}$.

Any particular sequence in the support of a channel describes one possible outcome of all communications from Alice to Bob - the list of all packets that Alice's computer sends to Bob's. The process of drawing from the channel, which results in a *sequence* of documents, is equivalent to a process that repeatedly draws a single "next" document from a distribution consistent with the history of already drawn documents - for example, drawing only packets which have a sequence number that is one greater than the sequence number of the previous packet. Therefore, we can think of communication as a series of these partial draws from the channel distribution, conditioned on what has been drawn so far. Notice that this notion of a channel is more general than the typical setting in which every symbol is drawn independently according to some fixed distribution: our channel explicitly models the dependence between symbols common in typical real-world communications.

Let $\mathcal{C}$ be a channel. We let $\mathcal{C}_h$ denote the marginal channel distribution on a single document from D conditioned on the history h of already drawn documents; we let $\mathcal{C}_h^l$ denote the marginal distribution on sequences of l documents conditioned on h. Concretely, for any $d \in D$, we will say that

$$\Pr_{\mathcal{C}_h}[d] = \frac{\sum_{s \in \{(h,d)\} \times D^*} \Pr_{\mathcal{C}}[s]}{\sum_{s \in \{h\} \times D^*} \Pr_{\mathcal{C}}[s]} \ ,$$

and that for any $\vec{d} \in d^l$,

$$\Pr_{\mathcal{C}_h^l}[\vec{d}] = \frac{\sum_{s \in \{(h,d)\} \times D^*} \Pr_{\mathcal{C}}[s]}{\sum_{s \in \{h\} \times D^*} \Pr_{\mathcal{C}}[s]} \ .$$

When we write "sample $x \leftarrow \mathcal{C}_h$" we mean that a single document should be returned according to the distribution conditioned on h. When it is not clear from context, we will use $\mathcal{C}_{A \to B,h}$ to denote the channel distribution on the communication from party A to party B.

Informativeness

We will require that a channel satisfy a minimum entropy constraint for all histories. Specifically, we require that there exist constants $L > 0$, $\beta > 0$, $\alpha > 0$ such that for all $h \in D^L$, either $\Pr_{\mathcal{C}}[h] = 0$ or $H_\infty(\mathcal{C}_h^\beta) \geq \alpha$. If a channel does not satisfy this property, then it is possible for Alice to drive the information content of her communications to 0, so this is a reasonable requirement. We say that a channel satisfying this condition is (L, α, β)-*informative*, and if a channel is (L, α, β)-informative for all $L > 0$, we say it is (α, β)-*always informative*, or simply *always informative*. Note that this definition implies an additive-like property of minimum entropy for marginal distributions, specifically, $H_\infty(\mathcal{C}_h^{l\beta}) \geq l\alpha$. For ease of exposition, we will assume channels are always informative in the remainder of this dissertation; however, our theorems easily extend to situations in which a channel is L-informative. The only complication in this situation is that there will be a bound in terms of (L, α, β) on the number of bits of secret message which can be hidden before the channel runs out of information.

Intuitively, L-informativeness requires that Alice always sends at least L non-null packets over her TCP connection to Bob, and at least one out of every β packets she sends has some probable alternative. Thus, we are requiring that Alice always says at least L/β "interesting things" to Bob.

Channel Access

In a multiparty setting, each ordered pair of parties (P, Q) will have their own channel distribution $\mathcal{C}_{P \to Q}$. To demonstrate that it is feasible to construct secure protocols for steganography, we will assume that party A has oracle access to marginal channel distributions $\mathcal{C}_{A \to B, h}$ for every other party B and history h. This is reasonable, because if Alice can communicate innocently with Bob *at all*, she must be able to draw from this distribution; thus we are only requiring that when using steganography, Alice can "pretend" she is communicating innocently.

On the other hand, we will assume that the adversary, Ward, knows as much as possible about the distribution on innocent communications. Thus he will be allowed oracle access to marginal channel distributions $\mathcal{C}_{P \to Q, h}$ for every pair P, Q and every history h. In addition, the adversary may be allowed access to an oracle which on input $(d, h, l) \in D^*$, returns an l-bit representation of $\Pr_{\mathcal{C}_h}[d]$.

These assumptions allow the adversary to learn as much as possible about any channel distribution but do not require any legitimate participant to know the distribution on communications from any other participant. We will, however, assume that each party knows (a summary of) the history of communications it has sent and received from every other participant; thus Bob must remember some details about the entire sequence of packets Alice sends to him.

Etc...

We will also assume that cryptographic primitives remain secure with respect to oracles which draw from the marginal channel distributions $\mathcal{C}_{A \to B,h}$. Thus channels which can be used to solve the hard problems that standard primitives are based on must be ruled out. In practice this is of little concern, since the existence of such channels would have previously led to the conclusion that the primitive in question was insecure.

Notice that the set of documents need not be literally interpreted as a set of bitstrings to be sent over a network. In general, documents could encode any kind of information, including things like actions – such as accessing a hard drive, or changing the color of a pixel – and times – such as pausing an extra $\frac{1}{2}$ second between words of a speech. In the single-party case, our theory is general enough to deal with these situations without any special treatment.

2.4 Bidirectional Channels: modeling interaction

Some of our protocols require an even more general definition of communications, to account for the differences in communications caused by interaction. For example, if Alice is a web browser and Bob is a web server, Alice's packets will depend on the packets she gets from Bob: if Bob sends Alice a web page with links to a picture, then Alice will also send Bob a request for that picture; and Alice's next request might more likely be a page linked from the page she is currently viewing. To model this interactive effect on communications, we will need a slightly augmented model. The main difference is that this channel is shared among two participants and messages sent by each participant might depend on previous messages sent by either one of them. To emphasize this difference, we use the term *bidirectional channel*.

Messages are still drawn from a set D of *documents*. For simplicity we assume that time proceeds in discrete *timesteps*. Each party $P \in \{P_0, P_1\}$ maintains a history h_P, which represents a timestep-ordered list of all documents sent and received by P. We call the set of well-formed histories $\mathcal{H}$. We associate to each party P a family of probability distributions $\mathcal{C}^P = \{\mathcal{C}_h^P\}_{h \in \mathcal{H}}$ on D.

The communication over a bidirectional channel $\mathcal{B} = (D, \mathcal{H}, \mathcal{C}^{P_0}, \mathcal{C}^{P_1})$ proceeds as follows. At each timestep, each party P receives messages sent to them in the previous timestep, updates h_P accordingly, and draws a document $d \leftarrow \mathcal{C}_{h_P}^P$ (the draw could result in the empty message $\perp$, signifying that no action should be taken that timestep). The document d is then sent to the other party and h_P is updated. We assume for simplicity that all messages sent at a given timestep are received at the next one. Denote by $\mathcal{C}_{h_P}^P \neq \perp$ the distribution $\mathcal{C}_{h_P}^P$ conditioned on not drawing $\perp$. We will consider families of bidirectional channels $\{\mathcal{B}_k\}_{k \geq 0}$ such that: (1) the length of elements in D_k is polynomially-bounded in k; (2) for each $h \in \mathcal{H}_k$ and party P, either $\Pr[\mathcal{C}_h^P = \perp] = 1$ or $\Pr[\mathcal{C}_h^P = \perp] \leq 1 - \delta$, for constant δ; and (3) there exists a function $\ell(k) = \omega(\log k)$ so that for each $h \in \mathcal{H}_k$, $H_\infty((\mathcal{C}_h^P)_k \neq \perp) \geq \ell(k)$ (that is, there is some variability in the communications).

Alternatively, a bi-directional channel can be thought of as a distribution on infinite sequences of *pairs* from $D' \times D'$, where $D' = D \cup \{\perp\}$, and the marginal distributions are distributions on the individual documents in a pair.

We assume that party P can draw from $\mathcal{C}_h^P$ for any history h, and that the adversary can draw from $\mathcal{C}_h^P$ for every party P and history h. We assume that the ability to draw from these distributions does not contradict the cryptographic assumptions that our results are based on. In the rest of the dissertation, all interactive communications will be assumed to conform to the bidirectional channel structure: parties only communicate by sending documents from D to each other and parties not running a protocol communicate according to the distributions specified by $\mathcal{B}$. Parties running a protocol strive to communicate using sequences of documents that appear to come from $\mathcal{B}$. As a convention, when $\mathcal{B}$ is compared to another random variable, we mean a random variable which draws from the process $\mathcal{B}$ the same number of documents as the variable we are comparing it to.

Bidirectional channels provide a model of the distribution on communications between two parties and are general enough to express almost any form of communication between the parties.

Chapter 3

Symmetric-key Steganography

Symmetric-key steganography is the most basic setting for steganography: Alice and Bob possess a shared secret key and would like to use it to exchange hidden messages over a public channel so that Ward cannot detect the presence of these messages. Despite the apparent simplicity of this scenario, there has been little work on giving a precise formulation of steganographic security. Our goal is to give such a formal description.

In Section 3.1, we give definitions dealing with the correctness and security of symmetric-key steganography. Then we show in Section 3.2 that these notions are *feasible* by giving constructions which satisfy them, under the assumption that pseudorandom function families exist. Finally, in section 3.3, we explore the *necessary* conditions for the existence of secure symmetric-key steganography.

3.1 Definitions

We will first define a stegosystem in terms of syntax and correctness, and then proceed to a security definition.

Definition 3.1. (Stegosystem) A steganographic protocol $\mathcal{S}$, or stegosystem, is a pair of probabilistic algorithms:

- $\mathcal{S}.\texttt{Encode}$ (abbreviated SE) takes as input a key $K \in \{0,1\}^k$, a string $m \in$

$\{0,1\}^*$ (the *hiddentext*), and a message history h.

$SE(K, m, h)$ returns a sequence of documents $s_1||s_2||\ldots||s_l$ (the *stegotext*) from the support of $\mathcal{C}_h^l$.

- $\mathcal{S}$.`Decode` (abbreviated SD) takes as input a key K, a sequence of documents $s_1||s_2||\ldots||s_l$, and a message history h.

  $SD(K, s, h)$ returns a hiddentext $m \in \{0,1\}^*$.

3.1.1 Correctness

Of course, in order for a stegosystem to be useful, it must be *correct:* when using the same key and history, decoding should recover any encoded message, most of the time:

Definition 3.2. (Correctness) A stegosystem $\mathcal{S}$ is *correct* if for every polynomial $p(k)$, there exists a negligible function $\mu(k)$ such that SE and SD also satisfy the relationship:

$$\forall m \in \{0,1\}^{p(k)}, h \in D^* : \quad \Pr(SD(K, SE(K, m, h), h) = m) \geq 1 - \mu(k) \ ,$$

where the randomization is over the key K and any coin tosses of SE, SD, and the oracles accessed by SE,SD.

An equivalent approach is to require that for any single-bit message, decoding correctly recovers an encoded bit with probability bounded away from $\frac{1}{2}$. In this case, multiple encodings under independent keys can be combined with error-correcting codes to make the probability of single-bit decoding failure negligible in k (we take a similar approach in our feasibility result). If the probability of decoding failure for a single-bit message is a negligible function $\mu(k)$, then for any polynomial $p(k)$, a union bound is sufficient to show that the probability of decoding failure for $p(k)$-bit messages is at most $p(k)\mu(k)$, which is still negligible in k.

3.1.2 Security

Intuitively, what we would like to require is that no efficient warden can distinguish between *stegotexts* output by SE and *covertexts* drawn from the channel distribution $\mathcal{C}_h$. As we stated in Section 2.3, we will assume that W knows the distribution $\mathcal{C}_h$; we will also allow W to know the algorithms involved in $\mathcal{S}$ as well as the history h of Alice's communications to Bob. In addition, we will allow W to pick the hiddentexts that Alice will hide, if she is in fact producing stegotexts. Thus, W's only uncertainty is about the key K and the single bit denoting whether Alice's outputs are stegotexts or covertexts.

As with encryption schemes, we will model an attack against a stegosystem as a game played by a *passive* warden, W, who is allowed to know the details of $\mathcal{S}$ and the channel $\mathcal{C}$.

Definition 3.3. (Chosen Hiddentext Attack) In a chosen hiddentext attack, W is given access to a "mystery oracle" M which is chosen from one of the following distributions:

1. ST: The oracle ST has a uniformly chosen key $K \leftarrow U_k$ and responds to queries (m, h) with a StegoText drawn from $SE(K, m, h)$.
2. CT: The oracle CT has a uniformly chosen K as well, and responds to queries (m, h) with a CoverText of length $\ell = |SE(K, m, h)|$ drawn from $\mathcal{C}_h^\ell$.

$W^M(1^k)$ outputs a bit which represents its guess about the type of M.

We define W's advantage against a stegosystem $\mathcal{S}$ for channel $\mathcal{C}$ by

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C},W}(k) = \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| ,$$

where the probability is taken over the randomness of ST, CT, and W.

Define the insecurity of $\mathcal{S}$ with respect to channel $\mathcal{C}$ by

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

where $\mathcal{W}(t, q, l)$ denotes the set of all adversaries which make at most $q(k)$ queries totaling at most $l(k)$ bits (of hiddentext) and running in time at most $t(k)$.

Definition 3.4. (Steganographic secrecy) A Stegosystem $\mathcal{S}_k$ is called (t, q, l, ϵ) *steganographically secret against chosen hiddentext attack* for the channel $\mathcal{C}$ ((t, q, l, ϵ)-SS-CHA-$\mathcal{C}$) if $\mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t, q, l, k) \leq \epsilon$.

Definition 3.5. (Universal Steganographic Secrecy) A Stegosystem $\mathcal{S}$ is called (t, q, l, ϵ)*-universally steganographically secret against chosen hiddentext attack* ((t, q, l, ϵ)-USS-CHA) if it is (t, q, l, ϵ)-SS-CHA-$\mathcal{C}$ for every always-informative channel $\mathcal{C}$.

A stegosystem is called *universally steganographically secret* USS-CHA if for every channel $\mathcal{C}$ and for every PPT W, $\mathbf{Adv}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C},W}(k)$ is negligible in k.

Note that steganographic secrecy can be thought of roughly as encryption which is indistinguishable from *arbitrary* distributions $\mathcal{D}$.

3.2 Constructions

For our feasibility results, 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 [16]). 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 our model which merely makes a single draw on the channel distribution. Further, we believe 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.

It is also worth noting that we assume that a stegosystem has very little knowledge of the channel distribution — SE may only *sample* from an oracle according to the distribution. This is because in many cases the full distribution of the channel has never been characterized; for example, the oracle may be a human being, or a video camera focused on some complex scene. However, our definitions do not rule out encoding procedures which have more detailed knowledge of the channel distribution.

Sampling from $\mathcal{C}_h$ might not be trivial. In some cases the oracle for $\mathcal{C}_h$ might be a human, and in others a simple randomized program. We stress that it is important to minimize the use of such an oracle, because oracle queries can be extremely expensive.

In practice, this oracle is also the weakest point of all our constructions. We 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 our constructions. A real world Warden would use this to his advantage. We therefore stress the following cautionary remark: *our protocols will be shown to be secure under the assumption that the channel oracle is perfect.*

3.2.1 A Stateful Construction

Setup: We assume Alice and Bob share a channel and let $\mathcal{C}$ denote the channel distribution. We write $d \leftarrow \mathcal{C}_h$ to denote the action of sampling d from the marginal distribution $\mathcal{C}_h$ (via oracle access). We let $F_K(\cdot, \cdot)$ denote a pseudorandom function family indexed by $k = |K|$ key bits which maps documents to bits, i.e. $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}$. We let Alice and Bob share a secret key $K \in \{0,1\}^k$ and also a synchronized d bit counter N (which need not be secret). The following procedures allow Alice and Bob to encode and decode a single bit and to send it via their shared channel.

Construction 3.6. `OneBit` Stegosystem

Procedure `OneBit.Encode`:
Input: $K \in \{0,1\}^k$, $m \in \{0,1\}$, $h \in D^*$, N
for $i = 1 \ldots \ell$ do
    draw $d_i, d_i' \leftarrow \mathcal{C}_h$
    If $F_K(N, d_i) = m$ then
      set $s_i = d_i$
    else
      set $s_i = d_i'$
    set $h = (h, s_i)$, increment N
Output: $s_1, s_2, \ldots, s_\ell$
	Procedure `OneBit.Decode`:
	Input: $K \in \{0,1\}^k$, $s_1, \ldots, s_\ell$, N
	set $C = 0$
	for $i = 1 \ldots l$ do
	    set $C = C + F_K(N, s_i)$
	    increment N
	if $C > \ell/2$ then
	  set $m = 1$
	else
	  set $m = 0$
	Output: m

The idea behind this construction is simple. The encoding algorithm makes ℓ copies of the bit m. For the i^{th} copy, the encoder attempts to find a document d such that $F_K(i, d) = m$, by drawing $d \leftarrow \mathcal{C}_h$. If the encoder fails, it draws a second $d' \leftarrow \mathcal{C}_h$ and sends d' instead of d. The decoder recovers ℓ bits by applying F_K to each bit, and outputs 1 if the majority of the bits are 1, and 0 otherwise. Intuitively, this works because we expect each s_i to map to the bit m with probability about $\frac{3}{4}$, and so the probability of a decoding failure should be negligible in ℓ.

Lemma 3.7. Let $s_1, \ldots, s_\ell \leftarrow SE(K, m, h)$ and suppose F is a uniformly chosen function. Then for any $d \in D$,

$$\Pr[s_i = d] = \Pr_{\mathcal{C}_{h_i}}[d] \ .$$

Proof. The event $s_i = d$ happens exactly when $d_i = d$ and $F_K(N + i, d_i) = m$ or when $d_i' = d$ and $F_K(N + i, d_i) = 1 - m$. Because d_i and d_i' are drawn independently from $\mathcal{C}_{h_i}$ and independently of F_K, and the conditions on $F_K(N + i, \cdot)$ are mutually exclusive, we get:

$$\begin{aligned}
\Pr[s_i = d] &= \Pr[(F_K(N+i, d_i) = m \wedge d_i = d) \vee (F_K(N+i, d_i) = 1 - m \wedge d_i' = d)] \\
&= \Pr[F_K(N+i, d_i) = m \wedge d_i = d] + \Pr[F_K(N+i, d_i) = 1 - m \wedge d_i' = d] \\
&= \Pr[F_K(N+i, d_i) = m] \Pr[d_i = d] + \Pr[F_K(N+i, d_i) = 1 - m] \Pr[d_i' = d] \\
&= \frac{1}{2} \Pr_{\mathcal{C}_{h_i}}[d] + \frac{1}{2} \Pr_{\mathcal{C}_{h_i}}[d] \\
&= \Pr_{\mathcal{C}_{h_i}}[d]
\end{aligned}$$

□

Lemma 3.8. Let $s_1, \ldots, s_\ell \leftarrow SE(K, m, h)$, and suppose F is a uniformly chosen function. Then for any i,

$$\Pr[F_K(N+i, s_i) = m] = \frac{1}{2} + \frac{1}{4} \Pr_{d_0, d_1 \leftarrow \mathcal{C}_{h_i}}[d_0 \neq d_1]$$

Proof. Consider the two documents d_i, d_i' that SE draws in iteration i. It will be the case that $F_K(N + i, s_i) = m$ exactly when either $F_K(N + i, d_i) = m$, which happens with probability $\frac{1}{2}$, or when $F_K(N + i, d_i) = 1 - m$ and $F_K(N + i, d_i') = m$, which happens with probability $\frac{1}{4}$ when $d_i \neq d_i'$, and with probability 0 otherwise. The theorem applies for any i because the function $F_K(N + i, \cdot)$ is independent of $F_K(N + j, \cdot)$ for $i \neq j$ when F_K is uniformly chosen. □

Lemma 3.9. Suppose $\mathcal{C}$ is (α, β)-always informative and F is a uniformly chosen function. Then we have

$$\Pr_i[F_K(N+i, s_i) = m] \geq \frac{1}{2} + \frac{1}{4\beta}(1 - 2^{-\alpha/\beta})$$

Proof. Because $\mathcal{C}$ is (α, β)-informative, for any h and any sequence $d_1, \ldots, d_\beta \leftarrow \mathcal{C}_h^\beta$, there must be a j between 0 and $\beta - 1$ such that $H_\infty(\mathcal{C}_{(h,ldots,d_j)}) \geq \alpha/\beta$. If this were not the case, then we would have h such that $H_\infty(\mathcal{C}_h^\beta) < \alpha$. Thus for a string of length ℓ drawn from $\mathcal{C}_h^\ell$, there must be ℓ/β positions i which have $H_\infty(\mathcal{C}_{h_i}) \geq \alpha/\beta$. In these positions, the collision probability is at most $2^{-\alpha/\beta}$. In the other positions, the collision probability is at most 1. Applying the previous lemma yields the result. $\square$

Theorem 3.10. *If $\mathcal{C}$ is (α, β)-informative, then*

$$\Pr[SD(K, SE(K, m, h), h) \neq m] \leq e^{-\gamma\ell} + \mathbf{InSec}_F^{\mathsf{prf}}(T_{SE}\ell, 2\ell, k) \ ,$$

where $\gamma = 2(\frac{1}{4\beta}(1 - 2^{-\alpha/\beta}))^2$ and T_{SE} is the time required to execute the inner loop of `OneBit.Encode`.

Proof. Lemma 3.9 implies that if F_K is a random function, then

$$\Pr[SD(K, SE(K, m, h), h) \neq m] \leq e^{-\gamma\ell} \ .$$

We describe a PRF-adversary A for F that has advantage

$$\left|\Pr[SD(K, SE(K, m, h), h) \neq m] - e^{-\gamma\ell}\right| \ .$$

A uses its function oracle f to emulate the action of SE encoding a uniformly chosen bit m under history h, counting the number of documents with $f(N+i, s_i) = m$. If fewer than $\frac{1}{2}$ of the s_i satisfy $f(N+i, s_i) = m$, A outputs 1, otherwise A outputs 0. Lemma 3.9 shows that $\Pr[A^f(1^k) = 1] \leq e^{-\gamma\ell}$, whereas

$$\Pr[A^{F_k}(1^k) = 1] = \Pr[SD(K, SE(K, m, h), h) \neq m] \ .$$

So by definition of advantage,

$$\mathbf{Adv}_{A,F}^{\mathsf{prf}}(k) \geq \left|\Pr[SD(K, SE(K, m, h), h) \neq m] - e^{-\gamma\ell}\right| \ ,$$

and it follows that this quantity is at most $\mathbf{InSec}_F^{\mathsf{prf}}(TIME(A), QUERIES(A), k)$. But A runs in time ℓT_{SE} and makes 2ℓ function-oracle queries, which proves the theorem. $\square$

Extending to multiple-bit messages

For completeness, we now state the obvious extension of the stegosystem `OneBit` to multiple-bit hiddentexts. We assume the same setup as previously.

Construction 3.11. `MultiBit` Stegosystem

Procedure `MultiBit.Encode`:	Procedure `MultiBit.Decode`:
Input: $K \in \{0,1\}^k$, $m \in \{0,1\}^L$, $h \in D^*$, N	**Input:** $K \in \{0,1\}^k$, $s_1, \ldots, s_{L\ell}$, N
for $i = 1 \ldots L$ do	for $i = 1 \ldots L$ do
draw $s_i \leftarrow \mathtt{OneBit.Encode}(K, m_i, h, N)$	set $S_i = s_{(i-1)\ell}, \ldots, s_{i\ell-1}$
set $h = (h, s_i)$, $N = N + \ell$	set $m_i = \mathtt{OneBit.Decode}(K, S_i, N)$
Output: $s_1, s_2, \ldots, s_{\lvert m \rvert}$	set $N = N + \ell$
	Output: $m_1 \| \cdots \| m_L$

The `MultiBit` stegosystem works by simply repeatedly invoking `OneBit` on the individual bits of the message m.

Theorem 3.12. *If $\mathcal{C}$ is (α, β)-informative, then*

$$\Pr[SD(K, SE(K, m, h, N), N) \neq m] \leq |m|(e^{-\gamma\ell}) + \mathbf{InSec}_F^{\mathsf{prf}}(|m|T_{SE}\ell, 2|m|\ell, k)),$$

where $\gamma = 2(\frac{1}{4\beta}(1 - 2^{-\alpha/\beta}))^2$ and T_{SE} is the time required to execute the inner loop of `OneBit.Encode`*.*

Proof. Because each s_i is generated using a different value of the counter N, each execution of the inner loop of `OneBit.Encode` is independent when called with a uniformly chosen function. Thus when a uniformly chosen function is used, executing `OneBit.Encode` $|m|$ times with different bits is the same as using $|m|$ independent keys, each with failure probability at most $e^{-\gamma\ell}$; a union bound shows that for a random function f, $\Pr[SD^f(SE^f(m, h, N), N) \neq m] \leq |m|(e^{-\gamma\ell})$. To complete the proof, we apply the same technique as in the proof of Theorem 3.10 □

We would like to make a security claim about the stegosystem `MultiBit`, but because the stegosystem does not fit our syntactic definition, we need a slightly modified version of the chosen-hiddentext attack game. We will modify the definition of the oracle distribution ST so that the oracle's private state will include the value N, initialized to 0 and properly incremented between queries. With this modified game in mind, we can state our theorem about the security of `MultiBit`:

Theorem 3.13. *Let $k = |K|$. For any $l \leq 2^d$:*

$$\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C}}(t,q,\mu,k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + \ell\mu T_{SE}, 2\ell\mu, k)$$

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

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C},W}(k) = \mathbf{Adv}^{\mathsf{prf}}_{F,A}(k)$$

The running time of A is the running time of warden W plus the time to make $\ell\mu$ passes through the inner loop of `OneBit.Encode`, or $\ell\mu T_{SE}$. The number of samples taken from $\mathcal{C}$ is at most $2\ell\mu$.

A^f simply runs W, playing the role of the oracle M in the chosen-hiddentext attack game. To respond to W's queries, A emulates the encoding procedure `MultiBit.Encode` using the function oracle f in place of $F_K(\cdot)$; A outputs the same bit as W. We consider the two cases for the oracle f:

- When f is a uniformly chosen function, then by lemma 3.7, the documents output by `OneBit.Encode` are distributed exactly according to $\mathcal{C}$. So

$$\Pr[A^f(1^k) = 1] = \Pr[W^{CT}(1^k) = 1] \ .$$

- When f is chosen from $F_K(\cdot,\cdot)$, the documents submitted to W are distributed identically to the output of `MultiBit`, by the definition of the construction. Thus

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1] \ .$$

So the advantage of A is:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \left|\Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C},W}(k)
\end{aligned}$$

And the theorem follows by the definition of insecurity. $\square$

Corollary 3.14. *If $F_K(\cdot,\cdot)$ is pseudorandom then* `MultiBit` *is universally steganographically secret against chosen-hiddentext attacks.*

Proof. If F is pseudorandom then for all PPT adversaries A, $\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k)$ is negligible in k. The definition of insecurity and Theorem 3.13 imply that for any cover channel, $\mathcal{C}$, the advantage of a warden will be negligible in k. This, in turn, implies the corollary. □

Removing the need for state

Having extended our construction to use multiple-bit messages, we can now remove the requirement for Alice and Bob to share a synchronized counter N. This construction will utilize the same setup as the previous constructions, except that Alice and Bob now share a second key $\kappa \in \{0,1\}^k$ to a pseudorandom function $G : \{0,1\}^k \times D^k \to \{0,1\}^{d/2}$.

Construction 3.15. `NoState` Stegosystem

Procedure `NoState.Encode`:	**Procedure** `NoState.Decode`:
Input: $K, \kappa \in \{0,1\}^k$, $m \in \{0,1\}^L$, $h \in D^*$	**Input:** $K, \kappa \in \{0,1\}^k$, S_1, S_2
$S_1 \leftarrow \mathcal{C}_h^k$	$N = 2^{d/2} G_\kappa(S_1)$
$N = 2^{d/2} G_\kappa(S_1)$	$m = \mathtt{MultiBit.Decode}(\mathtt{K}, \mathtt{S_2}, \mathtt{N})$
$S_2 \leftarrow \mathtt{MultiBit.Encode}(K, m, (h, S_1), N)$	**Output:** m
Output: S_1, S_2	

The `NoState` stegosystem works by choosing a long sequence from $\mathcal{C}_h$ (long enough that it is unlikely to repeat in the chosen-hiddentext attack game) and uses it to derive a value N, which is then used as the state for the `MultiBit` stegosystem. This value is always a multiple of $2^{d/2}$, so that if the value derived from the long sequence never repeats, then any messages of length at most $2^{d/2}$ will never use a value of N used by another message.

Theorem 3.16. *If $\mathcal{C}$ is (α, β)-informative, then*

$$\Pr[SD(K, SE(K, m, h)) \neq m] \leq |m|(e^{-\gamma\ell}) + \mathbf{InSec}^{\mathsf{prf}}_F(|m|T_{SE}\ell, 2|m|\ell, k)),$$

where $\gamma = 2(\frac{1}{4\beta}(1 - 2^{-\alpha/\beta}))^2$ and T_{SE} is the time required to execute the inner loop of `OneBit.Encode`.

Proof. The theorem follows directly from Theorem 3.12 $\square$

Theorem 3.17. *If $\mathcal{C}$ is (α, β)-informative, then for any $q, \mu \leq 2^{d/2}$:*

$$\begin{aligned}\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{NoState},\mathcal{C}}(t, q, \mu, k) \leq\ & \mathbf{InSec}^{\mathsf{prf}}_{F}(t + qT_G + \ell\mu T_{SE}, 2\ell\mu, k) \\ & + \mathbf{InSec}^{\mathsf{prf}}_{G}(t + \ell\mu, q, k) \\ & + \frac{q(q-1)}{2}(2^{-d/2} + 2^{-\alpha k/\beta})\end{aligned}$$

Proof. We reformulate the CT oracle in the chosen-hiddentext attack game so that the oracle has a key $\kappa \leftarrow U_k$ and evaluates G_κ on the first k documents of its reply (S, T) to every query. Let NC denote the event that the values $G_\kappa(S_1), \ldots, G_\kappa(S_q)$ are all distinct during the chosen-hiddentext attack game and let C denote the complement of NC.

Let W be any adversary in $\mathcal{W}(t, q, \mu)$, and assume without loss of generality that $\Pr[W^{ST}(1^k) = 1] > \Pr[W^{CT}(1^k) = 1]$. We wish to bound W's advantage against the stegosystem $\mathtt{NoState}$.

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{NoState},\mathcal{C},W}(k) &= \Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1] \\ &= \big(\Pr[W^{ST}(1^k) = 1|\mathsf{NC}]\Pr[\mathsf{NC}] + \Pr[W^{ST}(1^k) = 1|\mathsf{C}]\Pr[\mathsf{C}]\big) \\ &\quad - \big(\Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\Pr[\mathsf{NC}] + \Pr[W^{CT}(1^k) = 1|\mathsf{C}]\Pr[\mathsf{C}]\big) \\ &\leq \big(\Pr[W^{ST}(1^k) = 1|\mathsf{NC}]\Pr[\mathsf{NC}] - \Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\Pr[\mathsf{NC}]\big) \\ &\quad + \Pr[\mathsf{C}] \\ &\leq \big|\Pr[W^{ST}(1^k) = 1|\mathsf{NC}] - \Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\big| + \Pr[\mathsf{C}]\end{aligned}$$

We will show that for any W we can define an adversary X such that

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C},X}(k) \geq \big|\Pr[W^{ST}(1^k) = 1|\mathsf{NC}] - \Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\big| \ .$$

X plays the nonce-respecting chosen hiddentext attack game against $\mathtt{MultiBit}$ by running W and emulating W's oracle. To do this, X picks a key $\kappa \leftarrow U_k$, and when W makes the query (m, h), X draws $S_1 \leftarrow \mathcal{C}_h^k$, and computes $N = 2^{d/2}G_\kappa(S_1)$. If N is the same as some previous nonce used by X, X halts and outputs 0. Otherwise, X queries its oracle on $(m, (h, S_1), N)$ to get a sequence S_2, and then responds to W with S_1, S_2. Notice that

$$\Pr[X^{ST}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1|\mathsf{NC}] \ ,$$

and likewise that

$$\Pr[X^{CT}(1^k) = 1] = \Pr[W^{CT}(1^k) = 1|\mathsf{NC}] \ .$$

Thus we have that

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C},X}(k) = \left|\Pr[W^{ST}(1^k) = 1|\mathsf{NC}] - \Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\right| \ ,$$

and since X makes as many queries (of the same length) as W and runs in time $t + qT_G$, we have that

$$\begin{aligned}\left|\Pr[W^{ST}(1^k) = 1|\mathsf{NC}] - \Pr[W^{CT}(1^k) = 1|\mathsf{NC}]\right| &\leq \mathbf{InSec}^{\mathsf{ss}}_{\mathtt{MultiBit},\mathcal{C}}(t + qT_G, q, \mu) \\ &\leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + qT_G + \ell\mu T_{SE}, 2\ell\mu, k)\end{aligned}$$

by Theorem 3.13. Thus we need only to bound the term $\Pr[\mathsf{C}]$.

Consider a game played with the warden W in which a random function f is used in place of the function G_κ, and let C_f denote the same event as C in the previous game. Let $S_1, \ldots, S_q$ denote the k-document prefixes of the sequences returned by the oracle in the chosen-hiddentext attack game and let $N_i = f(S_i)$. Then the event C_f happens when there exist $i \neq j$ such that $N_i = N_j$, or equivalently $f(S_i) = f(S_j)$; and this event happens when $S_i = S_j$ or $S_i \neq S_j \wedge f(S_i) = f(S_j)$. Thus for a random f,

$$\begin{aligned}\Pr[\mathsf{C}_f] &= \Pr[\bigvee_{i<j<q} ((S_i = S_j) \vee (S_i \neq S_j \wedge f(S_i) = f(S_j)))] \\ &\leq \sum_{i<j<q} \Pr[S_i = S_j] + \Pr[f(S_i) = f(S_j) \wedge (S_i \neq S_j)] \\ &\leq \frac{q(q-1)}{2}\left(\Pr[S_i = S_j] + 2^{-d/2}\right) \\ &\leq \frac{q(q-1)}{2}\left(2^{-\alpha k/\beta} + 2^{-d/2}\right)\end{aligned}$$

Finally, observe that for every $W \in \mathcal{W}(t, q, \mu)$ we can construct a PRF-Adversary A for G in $\mathcal{A}(t + \ell\mu, q)$ such that

$$\mathbf{Adv}^{\mathsf{prf}}_{G,A}(k) \geq |\Pr[\mathsf{C}] - \Pr[\mathsf{C}_f]| \ .$$

A runs W, using its oracle f in place of G_κ to respond to W's queries. A outputs 1 if the event C_f occurs, and 0 otherwise. Notice that $\Pr[A^{G_\kappa}(1^k) = 1] = \Pr[\mathsf{C}]$ and

$\Pr[A^f(1^k) = 1] = \Pr[\mathsf{C}_f]$, which satisfies the claim. So to complete the proof, we observe that

$$
\begin{aligned}
\Pr[\mathsf{C}] &\leq |\Pr[\mathsf{C}] - \Pr[\mathsf{C}_\mathsf{f}]| + \Pr[\mathsf{C}_f] \\
&\leq \mathbf{InSec}^{\mathsf{prf}}_G(t + \ell\mu, q, k) + \Pr[\mathsf{C}_f] \\
&\leq \mathbf{InSec}^{\mathsf{prf}}_G(t + \ell\mu, q, k) + \frac{q(q-1)}{2}\left(2^{-\alpha k/\beta} + 2^{-d/2}\right)
\end{aligned}
$$

□

3.2.2 An Alternative Construction

The following protocol also satisfies our definition for universal steganographic secrecy. This protocol (up to small differences) is not new and can be found in [6]; an information theoretic version of the protocol can also be found in [16].

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) [54]. Suppose Alice and Bob share a key $K \in \{0,1\}^k$, and a function f such that $\Delta(f(\mathcal{C}_h), U_1) \leq \epsilon$ for any h. One example of such a function would be a uniformly chosen element of a universal hash family mapping $D^k \to \{0,1\}$; then when $\mathcal{C}$ is (α,β)-informative, we would have $\epsilon \leq 2^{1-\Omega(\alpha/2\beta)}$ except with negligible probability. The following procedures allow encoding and decoding of messages in a manner which is steganographically secret under chosen hiddentext attack for the channel distribution $\mathcal{C}$.

Construction 3.18. `UHash` Stegosystem

Procedure `UHash.Encode`:
Input: key K, hiddentext m, history h
Let $c = E_K(m)$
Parse c as $c_1||_1 c_2||_1 \cdots ||_1 c_\ell$
for $i = 1 \ldots \ell$ do
  set $j = 0$
  repeat:
    increment j
    $s_i \leftarrow \mathcal{C}_h$
  until $f(s_i) = c_i$ or $j = |K|$
  set $h = (h, s_i)$
Output: $s_1, s_2, \ldots, s_\ell$

Procedure `UHash.Decode`:
Input: key K, Stegotext $s_1, \ldots, s_\ell$
for $i = 1 \ldots l$ do
  set $c_i = f(s_i)$
set $c = c_1 \| c_2 \| \cdots \| c_l$.
Output: $D_K(c)$

The idea behind this construction is also simple. First, the encoding algorithm encrypts the hiddentext m to get a ciphertext $c = E_K(m)$. Next, is splits the ciphertext c into single bits. Then it draws documents from the channel until it finds s_1 such that $f(s_1) = c_1$. After that it finds s_2 such that $f(s_2) = c_2$, and s_3, and so on. This continues until all bits of c have been encoded. The decoding algorithm simply applies f to all the s_i's to get c and then decrypts with D_K to recover the plaintext. 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 our assumption of an "unbiased" function implies that an error correcting code as in construction 3.15 is unnecessary: the probability of failure here is at most $(1/2 + \epsilon)^{|K|}$. In the following, we will let $\ell(q, l)$ denote the total number of bits of ciphertext returned by encrypting q plaintexts of total length l.

Lemma 3.19. For any warden $W \in \mathcal{W}(t, q, l)$, we can construct a corresponding adversary A where

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{UHash},\mathcal{C},W}(k) \leq \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k) + \ell(q, l)\epsilon \ .$$

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. We are given a program W for distinguishing between the output of construction 3.18 and the ordinary traffic of the channel, and an oracle for sampling blocks from that channel. We construct a program A which plays the CPA game — distinguishing an E_K oracle from a uniform \$ oracle — with nearly the same advantage as W. A simply runs W, responding to his queries by executing the encoding procedure `UHash.Encode` using its oracle in place of E_K. Consider the following two cases:

- $O(m) = E_K(m)$. Then the stegotexts output by the encoding procedure will be identically distributed to stegotexts resulting from the normal use of construction 3.18.
- $O(m) = \$(m)$ is chosen uniformly from strings of appropriate length. Then the stegotexts output by the encoding procedure will be $\ell(l, q)\epsilon$-statistically close to samples from $\mathcal{C}_h$. To see that this is so, imagine instead that the i^{th} bit of

the ciphertext, c_i was chosen so that $\Pr[c_i = 0] = \Pr[f(\mathcal{C}_{h_i}) = 0]$. In this case the the i^{th} stegotext will come from a distribution identical to $\mathcal{C}_{h_i}$. But since $\Delta(c_i, U_1) \leq \epsilon$, it must be the case that $\Delta(s_i, \mathcal{C}_{h_i}) \leq \epsilon$ as well, by proposition 2.4.

Thus A can simply use the decision of W to gain advantage close to that of W. More formally,

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k) &= \left|\Pr[A^{E_K}(1^k) = 1] - \Pr[A^{\$}(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[A^{\$}(1^k) = 1]\right| \\
&\geq \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| + \ell(q,l)\epsilon \\
&= \mathbf{Adv}^{\mathsf{ss}}_{\mathtt{UHash},\mathcal{C},W}(k) + \ell(q,l)\epsilon
\end{aligned}$$

□

Theorem 3.20. $\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{UHash},\mathcal{C}}(t,q,l,k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kl), q, l, k) + \ell(q,l)\epsilon.$

Proof. The theorem follows from Lemma 3.19 and the definition of insecurity. □

3.3 Necessary Conditions for Steganography

The previous section demonstrates that relative to an oracle for $\mathcal{C}$, the existence of one-way functions is sufficient for the existence of secure steganography. In this section we will explore weaker definitions of steganographic secrecy, and establish two results. First, one-way functions are necessary for steganography; thus, relative to a channel oracle, the existence of one-way functions and secure steganography are equivalent. Second, we will show that in the "standard model," without access to a channel oracle, the existence of a secure stegosystem implies the existence of a program which samples from $\mathcal{C}_h$; and thus in the standard model, secure steganography for $\mathcal{C}$ exists if and only if $\mathcal{C}_h$ is efficiently sampleable.

3.3.1 Steganography implies one-way functions

To strengthen our result, we develop the weaker notion of security against known-hiddentext attacks (KHA). In a (l, μ)-KHA attack against distribution $\mathcal{D}$, the adversary is given a history h of length l, a hiddentext drawn from $\mathcal{D}^{\mu}$, and a sequence of documents $s \in D^{|SE(K,m,h)|}$. The adversary's task is to decide whether $s \leftarrow \mathcal{C}_h$ or $s \leftarrow SE(K, m, h)$. We define the KHA-advantage of W by

$$\mathbf{Adv}^{\mathsf{kha}\text{-}\mathcal{D}}_{\mathcal{S},\mathcal{C},W}(k, l, \mu) = \left|\Pr[W(h, m, SE(K, m, h)) = 1] - \Pr[W(h, m, \mathcal{C}_h^{|SE(K,m,h)|}) = 1]\right|$$

and say that $\mathcal{S}$ is secure against known hiddentext attack with respect to $\mathcal{D}$ and $\mathcal{C}$ (SS-KHA-$\mathcal{D}$-$\mathcal{C}$) if for every PPT W, for all polynomially-bounded l, μ, $\mathbf{Adv}^{\mathsf{kha}\text{-}\mathcal{D}}_{\mathcal{S},\mathcal{C},W}(k, l(k), \mu(k))$ is negligible in k.

Thus a stegosystem is secure against known-hiddentext attack if given the history h, and a plaintext m, an adversary cannot distinguish (asymptotically) between a stegotext encoding m and a covertext of the appropriate length drawn from $\mathcal{C}_h$. We will show that one-way functions are necessary even for this much weaker notion of security. In order to do so, we will use the following results from [33]:

Definition 3.21. ([33], Definition 3.9) A polynomial-time computable function $f : \{0,1\}^k \to \{0,1\}^{\ell(k)}$ is called a *false entropy generator* if there exists a polynomial-time computable $g : \{0,1\}^{k'} \to \{0,1\}^{\ell(k)}$ such that:

1. $H_S(g(U_{k'})) > H_S(f(U_k))$, and
2. $f(U_k) \approx g(U_k')$

Thus, a function is a false entropy generator (FEG) if it's output is indistinguishable from a distribution with higher (Shannon) entropy. It is shown in [33] that if FEGs exist, then PRGs exist:

Theorem 3.22. *([33], Lemma 4.16) If there exists a false entropy generator, then there exists a pseudorandom generator*

Theorem 3.23. *If there is a stegosystem $\mathcal{S}$ which is SS-KHA-$\mathcal{D}$-$\mathcal{C}$ secure for some hiddentext distribution $\mathcal{D}$ and some channel $\mathcal{C}$, then there exists a pseudorandom generator, relative to an oracle for $\mathcal{C}$.*

Proof. We will show how to construct a false entropy generator from $\mathcal{S}.\texttt{Encode}$, which when combined with Proposition 3.22 will imply the result.

Consider the function f which draws a hiddentext m of length $|k|^2$ from $\mathcal{D}$, and outputs $(SE(K, m, \varepsilon), m)$. Likewise, consider the function g which draws a hiddentext m of length $|K|^2$ from $\mathcal{D}$ and has the output distribution $(\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|}, m)$. Because $\mathcal{S}$ is SS-KHA-$\mathcal{D}$-$\mathcal{C}$ secure, it must be the case that $f(U_k) \approx g(U_{k'})$. Thus f and g satisfy condition (1) from definition 3.21.

Now, consider $H_S(\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|})$ versus $H_S(SE(K, m, h))$ We must have one of three cases:

1. $H_S(\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|}) > H_S(SE(K, m, \varepsilon))$; in this case the program that samples from $\mathcal{C}_\varepsilon$ is a false entropy generator and we are done.

2. $H_S(\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|}) < H_S(SE(K, m, \varepsilon))$; in this case SE is a false entropy generator, and again we are done.

3. $H_S(\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|}) = H_S(SE(K, m, \varepsilon))$; In this case, we have that

$$H_S(m|\mathcal{C}_\varepsilon^{|SE(K,m,\varepsilon)|}) = |K|^2 H_S(\mathcal{D}) ,$$

   whereas

$$H_S(m|SE(K, m, \varepsilon)) \leq (1 + \nu)|K|$$

   for a negligible function ν. To see that this is the case, notice that $m = SD(K, SE(K, m, \varepsilon))$ and so is determined (up to a negligible probability) by K, and $H_S(K) = |K|$. Thus asymptotically, we have that $H_S(f(U_k)) > H_S(g(U_{k'}))$, and f is a false entropy generator relative to an oracle for $\mathcal{C}$.

□

Corollary 3.24. *Relative to an oracle for $\mathcal{C}$, secure steganography for $\mathcal{C}$ exists if and only if one-way functions exist.*

Proof. The corollary follows from Theorem 3.23 and the results of Section 3.2 and [33].

□

3.3.2 Sampleable Channels are necessary

We say that a channel $\mathcal{C}$ is *efficiently sampleable* if there exists an algorithm $\mathtt{C}$ such that for any polynomial time A, for any polynomial l,

$$\left| \Pr_{h \leftarrow \mathcal{C}_\varepsilon^{l(k)}}[A(1^k, \mathtt{C}(h, 1^k, U_k))] - \Pr_{h \leftarrow \mathcal{C}_\varepsilon^{l(k)}}[A(1^k, \mathcal{C}_h)] \right|$$

is negligible in k. Notice that for any efficiently sampleable channel $\mathcal{C}$, the results of the previous sections prove that secure steganography with respect to $\mathcal{C}$ exists if and only if one-way functions exist *in the standard model* - e.g., without assuming oracle access to the channel $\mathcal{C}$. Here we will introduce a very weak notion of security with respect to $\mathcal{C}$ and show that if secure steganography exists for $\mathcal{C}$ in the standard model, then $\mathcal{C}$ is efficiently sampleable.

A weaker attack yet than the KHA attack is the *Known Distribution Attack* game: In a l-KDA attack against distribution $\mathcal{D}$, the adversary is given a history h of length l, and a sequence of documents $s \in D^{|SE(K,\mathcal{D},h)|}$. The adversary's task is to decide whether $s \leftarrow \mathcal{C}_h$ or $s \leftarrow SE(K, \mathcal{D}, h)$. We define the KDA-advantage of W by

$$\mathbf{Adv}^{\mathsf{kda}\text{-}\mathcal{D}}_{\mathcal{S},\mathcal{C},W}(k, l) = \left| \Pr_{h \leftarrow \mathcal{C}_\varepsilon^l, m \leftarrow \mathcal{D}}[W(SE(K, m, h)) = 1] - \Pr_{h \leftarrow \mathcal{C}_\varepsilon^l}[W(\mathcal{C}_h^\ell) = 1] \right|$$

and say that $\mathcal{S}$ is secure against known distribution attack with respect to $\mathcal{D}$ and $\mathcal{C}$ (SS-KDA-$\mathcal{D}$-$\mathcal{C}$) if for every PPT W, for all polynomially-bounded l, $\mathbf{Adv}^{\mathsf{kda}\text{-}\mathcal{D}}_{\mathcal{S},\mathcal{C},W}(k, l(k))$ is negligible in k. This attack is weaker yet than a KHA attack in that the length of the hiddentext is shorter and the hiddentext is unknown to W.

Theorem 3.25. *If there exists an efficiently sampleable $\mathcal{D}$ such that there is a* SS-KDA-*$\mathcal{D}$-$\mathcal{C}$ secure stegosystem $\mathcal{S}$ in the standard model, then $\mathcal{C}$ is efficiently sampleable.*

Proof. Consider the program $\mathtt{C}_\mathcal{S}$ with the following behavior: on input $(1^k, h)$, $\mathtt{C}_\mathcal{S}$ picks $K \leftarrow \{0, 1\}^k$, picks $m \leftarrow \mathcal{D}$, and returns the first document of $\mathcal{S}.\mathtt{Encode}(K, m, h)$. Consider any PPT distinguisher A. We will that the KDA adversary W which passes the first document of its input to A and outputs A's decision has at least the advantage of A. This is because in case W's input is drawn from SE, the input it passes to A is exactly distributed according to $\mathtt{C}_\mathcal{S}(1^k, h)$; and when W's input is drawn from $\mathcal{C}_h$, the input it passes to A is exactly distributed according to $\mathcal{C}_h$:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{kda}\text{-}\mathcal{D}}_{\mathcal{S},\mathcal{C},\mathcal{W}}(k,|h|) &= |\Pr[W(SE(K,m,h))=1] - \Pr[W(\mathcal{C}_h)=1]| \\ &= \left|\Pr[A(1^k,\mathtt{C}_{\mathcal{S}}(1^k,h))=1] - \Pr[A(1^k,\mathcal{C}_h)=1]\right| \ .\end{aligned}$$

But because $\mathcal{S}$ is $\mathsf{SS\text{-}KDA}$-$\mathcal{D}$-$\mathcal{C}$ secure, we know that W's advantage must be negligible, and thus no efficient A can distinguish this from the first document drawn from $\mathcal{C}_h^{|SE(K,\mathcal{D},h)|}$. So the output of $\mathtt{CC}_{\mathcal{S}}$ is computationally indistinguishable from $\mathcal{C}$. $\square$

As a consequence of this theorem, if a designer is interested in developing a stegosystem for some channel $\mathcal{C}$ in the standard model, he can focus exclusively on designing an efficient sampling algorithm for $\mathcal{C}$. If his stegosystem is secure, it will include one anyway; and if he can design one, he can "plug it in" to the constructions from section 3.2 and get a secure stegosystem based on "standard" assumptions.

Chapter 4

Public-Key Steganography

The results of the previous chapter assume that the sender and receiver share a secret, randomly chosen key. In the case that some exchange of key material was possible before the use of steganography was necessary, this may be a reasonable assumption. In the more general case, two parties may wish to communicate steganographically, without prior agreement on a secret key. We call such communication *public key steganography*. Whereas previous work has shown that symmetric-key steganography is possible – though inefficient – in an information-theoretic model, public steganography is information-theoretically *impossible*. Thus our complexity-theoretic formulation of steganographic secrecy is crucial to the security of the constructions in this chapter.

In Section 4.1 we will introduce some required basic primitives from the theory of public-key cryptography. In Section 4.2 we will give definitions for public-key steganography and show how to use the primitives to construct a public-key stegosystem. Finally, in Section 4.3 we introduce the notion of steganographic key exchange and give a construction which is secure under the Integer Decisional Diffie-Hellman assumption.

4.1 Public key cryptography

Our results build on several well-established cryptographic assumptions from the theory of public-key cryptography. We will briefly review them here, for completeness.

Integer Decisional Diffie-Hellman.

Let P and Q be primes such that Q divides $P-1$, let $\mathbb{Z}_P^*$ be the multiplicative group of integers modulo P, and let $g \in \mathbb{Z}_P^*$ have order Q. Let A be an adversary that takes as input three elements of $\mathbb{Z}_P^*$ and outputs a single bit. Define the *DDH advantage of A over* (g, P, Q) as: $\mathbf{Adv}_A^{\mathsf{ddh}}(g, P, Q) = |\Pr_{a,b}[A(g^a, g^b, g^{ab}, g, P, Q) = 1] - \Pr_{a,b,c}[A(g^a, g^b, g^c, g, P, Q) = 1]|$, where a, b, c are chosen uniformly at random from $\mathbb{Z}_Q$ and all the multiplications are over $\mathbb{Z}_P^*$. The Integer Decisional Diffie-Hellman assumption (DDH) states that for every PPT A, for every sequence $\{(P_k, Q_k, g_k)\}_k$ satisfying $|P_k| = k$ and $|Q_k| = \Theta(k)$, $\mathbf{Adv}_A^{\mathsf{ddh}}(g_k, P_k, Q_k)$ is negligible in k.

Trapdoor One-way Permutations.

A trapdoor one-way permutation family Π is a sequence of sets $\{\Pi_k\}_k$, where each Π_k is a set of bijective functions $\pi : \{0,1\}^k \to \{0,1\}^k$, along with a triple of algorithms (G, E, I). $G(1^k)$ samples an element $\pi \in \Pi_k$ along with a *trapdoor* τ; $E(\pi, x)$ evaluates $\pi(x)$ for $x \in \{0,1\}^k$; and $I(\tau, y)$ evaluates $\pi^{-1}(y)$. For a PPT A running in time $t(k)$, denote the advantage of A against Π by

$$\mathbf{Adv}_{\Pi,A}^{\mathsf{ow}}(k) = \Pr_{(\pi,\tau)\leftarrow G(1^k), x \leftarrow U_k}[A(\pi(x)) = x] \ .$$

Define the insecurity of Π by $\mathbf{InSec}_{\Pi}^{\mathsf{ow}}(t, k) = \max_{A \in \mathcal{A}(t)} \left\{ \mathbf{Adv}_{\Pi,A}^{\mathsf{ow}}(k) \right\}$, where $\mathcal{A}(t)$ denotes the set of all adversaries running in time $t(k)$. We say that Π is a trapdoor one-way permutation family if for every probabilistic polynomial-time (PPT) A, $\mathbf{Adv}_{\Pi,A}^{\mathsf{ow}}(k)$ is negligible in k.

Trapdoor one-way predicates

A *trapdoor one-way predicate family* P is a sequence $\{P_k\}_k$, where each P_k is a set of efficiently computable predicates $p : D_p \to \{0,1\}$, along with an algorithm $G(1^k)$ that samples pairs (p, S_p) uniformly from P_k; S_p is an algorithm that, on input $b \in \{0,1\}$ samples x uniformly from D_p subject to $p(x) = b$. For a PPT A running in time $t(k)$, denote the advantage of A against P by

$$\mathbf{Adv}^{\mathsf{tp}}_{P,A}(k) = \Pr_{(p,S_p)\leftarrow G(1^k),x\leftarrow D_p}[A(x, S_p) = p(x)] \ .$$

Define the insecurity of P by

$$\mathbf{InSec}^{\mathsf{tp}}_{P}(t,k) = \max_{A\in\mathcal{A}(t)} \left\{\mathbf{Adv}^{\mathsf{tp}}_{P,A}(k)\right\} \ ,$$

where $\mathcal{A}(t)$ denotes the set of all adversaries running in time $t(k)$. We say that P is a trapdoor one-way predicate family if for every probabilistic polynomial-time (PPT) A, $\mathbf{Adv}^{\mathsf{tp}}_{P,A}(k)$ is negligible in k.

Notice that one way to construct a trapdoor one-way predicate is to utilize the Goldreich-Levin hard-core bit [28] of a trapdoor one-way permutation. That is, for a permutation family Π, the associated trapdoor predicate family P_Π works as follows: the predicate p_π has domain $Dom(\pi) \times \{0,1\}^k$, and is defined by $p(x,r) = \pi^{-1}(x) \cdot r$, where $\cdot$ denotes the vector inner product on $GF(2)^k$. [28] prove that there exist polynomials such that $\mathbf{InSec}^{\mathsf{tp}}_{P_\pi}(t,k) \leq poly(\mathbf{InSec}^{\mathsf{ow}}_{\Pi}(poly(t),k))$.

4.1.1 Pseudorandom Public-Key Encryption

We will require public-key encryption schemes that are secure in a slightly non-standard model, which we will denote by IND\$-CPA in contrast to the more standard IND-CPA. The main difference is that security against IND\$-CPA requires the output of the encryption algorithm to be indistinguishable from uniformly chosen random bits, whereas IND-CPA only requires the output of the encryption algorithm to be indistinguishable from encryptions of other messages.

Formally, a public-key (or asymmetric) cryptosystem $\mathcal{E}$ consists of three (randomized) algorithms:

- $\mathcal{E}$.$\mathtt{Generate}$: $1^k \to \mathcal{PK}_k \times \mathcal{SK}_k$ generates (public, secret) key pairs (PK, SK). We will abbreviate $\mathcal{E}.\mathtt{Generate}(1^k)$ by $G(1^k)$, when it is clear which encryption scheme is meant.

- $\mathcal{E}$.$\mathtt{Encrypt}$: $\mathcal{PK} \times \{0,1\}^* \to \{0,1\}^*$ uses a public key to transform a *plaintext* into a *ciphertext*. We will abbreviate $\mathcal{E}.\mathtt{Encrypt}(PK, \cdot)$ by $E_{PK}(\cdot)$.

- $\mathcal{E}$.$\mathtt{Decrypt}$: $\mathcal{SK} \times \{0,1\}^* \to \{0,1\}^*$ uses a secret key to transform a ciphertext into the corresponding plaintext. We will abbreviate $\mathcal{E}.\mathtt{Decrypt}(SK, \cdot)$ by $D_{SK}(\cdot)$.

Such that for all key pairs $(PK, SK) \in G(1^k)$, $\mathtt{Decrypt}(SK, \mathtt{Encrypt}(PK, m)) = m$.

To formally define the security condition for a public-key encryption scheme, consider a game in which an adversary A is given a public key drawn from $G(1^k)$ and chooses a message m_A. Then A is given either $E_{PK}(m_A)$ or a uniformly chosen string of the same length. Let $\mathcal{A}(t, l)$ be the set of adversaries A which produce a message of length at most $l(k)$ bits and run for at most $t(k)$ time steps. Define the IND\$-CPA advantage of A against $\mathcal{E}$ as

$$\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k) = \left| \Pr_{PK}[A(PK, E_{PK}(m_A)) = 1] - \Pr_{PK}[A(PK, U_{|E_{PK}(m_A)|}) = 1] \right|$$

Define the insecurity of $\mathcal{E}$ as $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, l, k) = \max_{A \in \mathcal{A}(t,l)} \left\{ \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k) \right\}$. $\mathcal{E}$ is (t, l, k, ϵ) - *indistinguishable from random bits under chosen plaintext attack* if $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, l, k) \leq \epsilon(k)$. $\mathcal{E}$ is called *indistinguishable from random bits under chosen plaintext attack* (IND\$-CPA) if for every probabilistic polynomial-time (PPT) A, $\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k)$ is negligible in k. We show how to construct IND\$-CPA public-key encryption schemes from a variety of well-established cryptographic assumptions.

IND\$-CPA public-key encryption schemes can be constructed from any primitive which implies trapdoor one-way predicates p with domains D_p satisfying one of the following conditions:

- D_p is computationally or statistically indistinguishable from $\{0,1\}^{poly(k)}$: in this case it follows directly that encrypting the bit b by sampling from $p^{-1}(b)$ yields an IND\$-CPA scheme. The results of Goldreich and Levin imply that such predicates exist if there exist trapdoor one-way permutations on $\{0,1\}^k$, for example.

- D_p has an efficiently recognizable, polynomially dense encoding in $\{0,1\}^{poly(k)}$; in this case, we let $q(\cdot)$ denote the polynomial such that every D_p has density at least $1/q(k)$. Then to encrypt a bit b, we draw $\ell = kq(k)$ samples $d_1, \dots, d_\ell \leftarrow U_{poly(k)}$; let i be the least i such that $d_i \in D_p$; then transmit $d_1, \dots, d_{i-1}, p^{-1}(b), d_{i+1}, \dots, d_\ell$. (This assumption is similar to the requirement for common-domain trapdoor systems used by [19], and all (publicly-known) public-key encryption systems seem to support construction of trapdoor predicates satisfying this condition.)

Stronger assumptions allow construction of more efficient schemes. Here we will construct schemes satisfying IND\$-CPA under the following assumptions: trapdoor one-way permutations on $\{0,1\}^k$, the RSA assumption, and the Decisional Diffie-Hellman assumption. Notice that although both of the latter two assumptions imply the former through standard constructions, the standard constructions exhibit considerable security loss which can be avoided by our direct constructions.

4.1.2 Efficient Probabilistic Encryption

The following "EPE" encryption scheme is described in [30], and is a generalization of the protocol given by [13]. When used in conjunction with a family of trapdoor one-way permutations on domain $\{0,1\}^k$, it is easy to see that the scheme satisfies IND\$-CPA:

Construction 4.1. (EPE Encryption Scheme)

Procedure `Encrypt`:	**Procedure** `Decrypt`:
Input: $m \in \{0,1\}^*$, tOWP π	**Input:** (x, r, c), trapdoor π^{-1}
Sample $x_0, r \leftarrow U_k$	let $l = \|c\|$, $x_l = x$
let $l = \|m\|$	for $i = l \dots 1$ do
for $i = 1 \dots l$ do	set $x_{i-1} = \pi^{-1}(x_i)$
set $b_i = x_{i-1} \odot r$	set $b_i = x_{i-1} \odot r$
set $x_i = f(x_{i-1})$	**Output:** $c \oplus b$
Output: $x_l, r, b \oplus m$	

IND\$-CPA-ness follows by the pseudorandomness of the bit sequence $b_1, \ldots, b_l$ generated by the scheme and the fact that x_l is uniformly distributed in $\{0,1\}^k$.

RSA-based construction

The RSA function $E_{N,e}(x) = x^e \bmod N$ is believed to be a trapdoor one-way permutation family when N is selected as the product of two large, random primes. The following construction uses Young and Yung's Probabilistic Bias Removal Method (PBRM) [65] to remove the bias incurred by selecting an element from $\mathbb{Z}_N^*$ rather than U_k.

Construction 4.2. (RSA-based Pseudorandom Encryption Scheme)

Procedure `Encrypt`:
Input: plaintext m; public key N, e
let $k = |N|$, $l = |m|$
repeat:
  Sample $x_0 \leftarrow \mathbb{Z}_N^*$
  for $i = 1 \ldots l$ do
    set $b_i = x_{i-1} \bmod 2$
    set $x_i = x_{i-1}^e \bmod N$
  sample $c \leftarrow U_1$
until $(x_l \leq 2^k - N)$ OR $c = 1$
if $(x_1 \leq 2^k - N)$ and $c = 0$ set $x' = x$
if $(x_1 \leq 2^k - N)$ and $c = 1$ set $x' = 2^k - x$
Output: $x', b \oplus m$

Procedure `Decrypt`:
Input: x', c; (N, d)
let $l = |c|$, $k = |N|$
if $(x' > N)$ set $x_l = x'$
else set $x_l = 2^k - x'$
for $i = l \ldots 1$ do
  set $x_{i-1} = x_i^d \bmod N$
  set $b_i = x_{i-1} \bmod 2$
Output: $c \oplus b$

The IND\$-CPA security of the scheme follows from the correctness of PBRM and the fact that the least-significant bit is a hardcore bit for RSA. Notice that the expected number of repeats in the encryption routine is at most 2.

DDH-based construction

Let $E_{(\cdot)}(\cdot)$, $D_{(\cdot)}(\cdot)$ denote the encryption and decryption functions of a *private-key* encryption scheme satisfying IND\$-CPA, keyed by κ-bit keys, and let $\kappa \leq k/3$. (We give an example of such a scheme in Chapter 2.) Let $\mathcal{H}_k$ be a family of pairwise-independent hash functions $H : \{0,1\}^k \to \{0,1\}^\kappa$. We let P be a k-bit prime (so

$2^{k-1} < P < 2^k$), and let $P = rQ + 1$ where $(r, Q) = 1$ and Q is also a prime. Let g generate $\mathbb{Z}_P^*$ and $\hat{g} = g^r \bmod P$ generate the unique subgroup of order Q. The security of the following scheme follows from the Decisional Diffie-Hellman assumption, the leftover-hash lemma, and the security of (E, D):

Construction 4.3. `DDHRand` Public-key cryptosystem.

Procedure `Encrypt`:	Procedure `Decrypt`:
Input: $m \in \{0,1\}^*$; $(g, \hat{g}^x, P)$	**Input:** (H, s, c); private key (x, P, Q)
Sample $H \leftarrow \mathcal{H}_k$	let $r = (P-1)/Q$
repeat:	set $K = H(s^{rx} \bmod P)$
Sample $y \leftarrow \mathbb{Z}_{P-1}$	**Output:** $D_K(c)$
until $(g^y \bmod P) \leq 2^{k-1}$	
set $K = H((\hat{g}^x)^y \bmod P)$	
Output: $H, g^y, E_K(m)$	

The security proof considers two hybrid encryption schemes: H_1 replaces the value $(\hat{g}^a)^b$ by a random element of the subgroup of order Q, $\hat{g}^c$, and H_2 replaces K by a random draw from $\{0,1\}^\kappa$. Clearly distinguishing H_2 from random bits requires distinguishing some $E_K(m)$ from random bits. The Leftover Hash Lemma gives that the statistical distance between H_2 and H_1 is at most $2^{-\kappa}$. Thus

$$\mathbf{Adv}_A^{H_1,\$}(k) \leq \mathbf{InSec}_E^{\mathsf{cpa}}(t, |\kappa|) + 2^{-\kappa} \ .$$

Finally, we show that any distinguisher A for H_1 from the output of `Encrypt` with advantage ϵ can be used to construct a distinguisher B that solves the DDH problem with advantage at least $\epsilon/2$. B takes as input a triple $(\hat{g}^x, \hat{g}^y, \hat{g}^z)$ and attempts to decide whether $z = xy$, as follows. First, B computes $\hat{r}$ as the least integer such that $r\hat{r} = 1 \bmod Q$, and then picks $\beta \leftarrow \mathbb{Z}_r$. Then B computes $s = (\hat{g}^y)^{\hat{r}} g^{\beta Q}$. If $s > 2^{k-1}$, B outputs 0. Otherwise, B submits $\hat{g}^x$ to A to get the message m_A, draws $H \leftarrow \mathcal{H}_k$, and outputs the decision of $A(\hat{g}^x, H \| s \| E_{H(\hat{g}^z)}(m_A))$. We claim that:

- The element s is a uniformly chosen element of $\mathbb{Z}_P^*$, when $y \leftarrow \mathbb{Z}_Q$. To see that this is true, observe that the exponent of s, $\xi = r\hat{r}y + \beta Q$, is congruent to $y \bmod Q$ and $\beta Q \bmod r$; and that for uniform β, βQ is also a uniform residue mod r. By the Chinese remainder theorem, there is exactly one element of $\mathbb{Z}_{rQ} =$

$\mathbb{Z}_{P-1}$ that satisfies these conditions, for every y and β. Thus s is uniformly chosen.

- B halts and outputs 0 with probability at most $\frac{1}{2}$ over input and random choices; and conditioned on not halting, the value s is uniformly distributed in $\{0,1\}^k$. This is true because $2^k/P < \frac{1}{2}$, by assumption.

- When $z = xy$, the input $H\|s\|E_{H(\hat{g}^z)}(m_A))$ is selected exactly according to the output of $\texttt{Encrypt}(\hat{g}^x, m_A)$. This is because

$$\begin{aligned}(\hat{g}^x)^\xi &= (g^{r\hat{r}y+\beta Q})^{rx} \\ &= g^{(\alpha Q+1)rxy+rQ(\beta x)} \\ &= g^{rxy} = \hat{g}^z\end{aligned}$$

- When $z \neq xy$, the input $H\|s\|E_{H(\hat{g}^z)}(m_A))$ is selected exactly according to the output of H_1, by construction.

Thus,

$$\Pr[B(\hat{g}^x, \hat{g}^y, \hat{g}^{xy}) = 1] = \frac{2^k}{P} \Pr[A(\hat{g}^x, \texttt{Encrypt}(\hat{g}^x, m_A)) = 1] ,$$

and

$$\Pr[B(\hat{g}^x, \hat{g}^y, \hat{g}^z) = 1] = \frac{2^k}{P} \Pr[A(\hat{g}^x, H_1(m_A)) = 1] .$$

And thus $\mathbf{Adv}^{\mathsf{ddh}}_B(\hat{g}, P, Q) \geq \frac{1}{2}\epsilon$. Thus, we have that overall,

$$\mathbf{InSec}^{\mathsf{cpa}}_{\texttt{DDHRand}}(t, l, k) \leq \mathbf{InSec}^{\mathsf{ddh}}_{g,P,Q}(t, k) + \mathbf{InSec}^{\mathsf{cpa}}_{(E,D)}t, l, 1, k + 2^{-\kappa} .$$

4.2 Public key steganography

We will first give definitions of public-key stegosystems and security against chosen-hiddentext attack, and then give a construction of a public-key stegosystem to demonstrate the feasibility of these notions. The construction is secure assuming the existence of a public-key IND\$-CPA-secure cryptosystem.

4.2.1 Public-key stegosystems

As with the symmetric case, we will first define a stegosystem in terms of syntax and correctness, and then proceed to a security definition.

Definition 4.4. (Stegosystem) A public-key stegosystem $\mathcal{S}$ is a triple of probabilistic algorithms:

- $\mathcal{S}$.`Generate` takes as input a security parameter 1^k and generates a key pair $(PK, SK) \in \mathcal{PK} \times \mathcal{SK}$. When it is clear from the context which stegosystem we are referring to, we will abbreviate $\mathcal{S}$.`Generate` by SG.

- $\mathcal{S}$.`Encode` (abbreviated SE when $\mathcal{S}$ is clear from the context) takes as input a public key $PK \in \mathcal{PK}$, a string $m \in \{0,1\}^*$ (the *hiddentext*), and a message history h. As with the symmetric case, we will also assume for our feasibility results that SE has access to a channel oracle for some channel $\mathcal{C}$, which can sample from $\mathcal{C}_h$ for any h.

  $SE(PK, m, h)$ returns a sequence of documents $s_1, s_2, \ldots, s_l$ (the *stegotext*) from the support of $\mathcal{C}_h^l$.

- $\mathcal{S}$.`Decode` (abbreviated SD) takes as input a secret key $SK \in \mathcal{SK}$, a sequence of documents $s_1, s_2, \ldots, s_l$, and a message history h.

  $SD(SK, s, h)$ returns a hiddentext $m \in \{0,1\}^*$.

As in the private key case, we will also require that a public-key stegosystem is correct:

Definition 4.5. (Correctness) A public-key stegosystem $\mathcal{S}$ is *correct* if for every polynomial $p(k)$ there exists a negligible $\nu(k)$ such that

$$\forall m \in \{0,1\}^{p(k)}, h \in D^* : \Pr_{(PK,SK)\leftarrow SG(1^k)}[SD(SK, SE(PK, m, h), h) = m] \geq 1 - \mu(k)$$

where the randomization is also over any coin tosses of SE, SD, SG and the oracle to $\mathcal{C}_h$.

Remark. While in general Alice will need to remember the history of documents transmitted to Bob, it is most desirable if Bob is not required to store the history of documents he has received from Alice. Some of our protocols require Bob to store this history, but it is straightforward to rewrite them so that Bob need only remember a collision-intractable digest of the history. In this paper we are not concerned with attacks in which Ward attempts to disrupt the communication between Alice and Bob, so the dependence of decoding on accurate history is chiefly a storage concern.

4.2.2 Steganographic Secrecy against Chosen Hiddentext Attack

In a chosen hiddentext attack, W is allowed to supply hiddentexts to Alice, in hopes of detecting whether or not she is using steganography in her communications with Bob. (Note that a chosen hiddentext attack is the steganographic equivalent of a chosen plaintext attack against an encryption scheme.)

We will model a warden attacking a stegosystem as an efficient oracle machine which plays the following game. First, a pair (PK, SK) is drawn from $SG(1^k)$ and W is allowed to access PK and sample from $\mathcal{C}_h$ for any history h, to produce a message m_W and a history h_W. Then W is given access to a sequence $s \in D^*$, which is drawn either from $SE(PK, m_W, h_W)$ or $\mathcal{C}_h^{|s|}$, and continues to access $\mathcal{C}_h$ for any history h (This oracle allows W to learn the *covertext* distribution on all communications). Finally W outputs a bit which is a "guess" about whether s is a stegotext or a covertext.

We define the *CHA advantage of W against stegosystem $\mathcal{S}$ over channel $\mathcal{C}$* by

$$\mathbf{Adv}^{\mathsf{cha}}_{\mathcal{S},\mathcal{C},W}(k) = \left| \Pr[W(PK, s) = 1] - \Pr[W(PK, \mathcal{C}_{h_W}^{|s|}) = 1] \right| \ ,$$

where the probabilities are taken over $(PK, SK) \leftarrow SG(1^k)$, $(m_W, h_W) \leftarrow W^(PK)$, $s \leftarrow SE(PK, m_W, h_W)$, the random bits of W and the responses of the oracle $\mathcal{C}$. Define *the insecurity of $\mathcal{S}$ over channel $\mathcal{C}$* by

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{S},\mathcal{C}}(t, l, k) = \max_{W \in \mathcal{W}(t,l)} \left\{ \mathbf{Adv}^{\mathsf{cha}}_{\mathcal{S},\mathcal{C},W}(k) \right\} \ ,$$

where $\mathcal{W}(t, l)$ denotes the set of all adversaries which choose a message of length at most $l(k)$ bits, and run in time at most $t(k)$.

Thus, $\mathcal{S}$ is secure against chosen-hiddentext attacks if the warden, even given Bob's public key, cannot tell whether Alice's message to Bob encodes any hiddentext at all, even one of his choice.

4.2.3 Construction

Our protocols build on the following construction, a generalization of construction 3.18. Let $f : D \to \{0, 1\}$ be a public function (recall that $\mathcal{C}$ is a distribution on sequences of elements of D). If f is is perfectly unbiased on $\mathcal{C}_h$ for all h, then the following encoding procedure, on uniformly distributed l-bit input c, produces output distributed exactly according to $\mathcal{C}_h^l$:

Construction 4.6. (Basic encoding/decoding routines)

Procedure `Basic_Encode`:
Input: bits $c_1, \ldots, c_l$, history h, bound k
for $i = 1 \ldots l$ do
  Let $j = 0$
  repeat:
    sample $s_i \leftarrow \mathcal{C}_h$, increment j
  until $f(s_i) = c_i$ OR $(j > k)$
  set $h = (h, s_i)$
Output: $s_1, s_2, \ldots, s_l$

Procedure `Basic_Decode`:
Input: Stegotext $s_1, s_2, \ldots, s_l$
for $i = 1 \ldots l$ do
  set $c_i = f(s_i)$
set $c = c_1||c_2|| \cdots ||c_l$.
Output: c

Note that for infinitely many $\mathcal{C}_h$ there is no perfectly unbiased function f. As with construction 3.18, this can be rectified by using a (global) universal hash function.

Lemma 4.7. Any channel $\mathcal{C}$ which is always informative can be compiled into a channel $\mathcal{C}^{(k)}$ which admits an efficiently computable function f such that for any polynomial-length sequence $h_1, \ldots, h_n$ satisfying $\Pr_{\mathcal{C}}[h_i] \neq 0$, $\left|\Pr[f(C_{h_i}^{(k)}) = 1] - \frac{1}{2}\right|$ is negligible in k for all $1 \leq i \leq n$.

Proof. Let $l(k) = \omega(\log k)$. Then the channel $\mathcal{C}^{(k)}$ is simply a distribution on sequences of documents which are elements of $D^{l(k)}$ and the marginal distributions $\mathcal{C}_h^{(k)}$ are simply $\mathcal{C}_h^{l(k)}$. Because $\mathcal{C}$ is always informative, we have that for any h which has non-zero probability, $H_\infty(\mathcal{C}_h^{(k)}) = \omega(\log k)$.

Let $h_1, h_2, ..., h_m$ be any sequence of histories which all have non-zero probability under $\mathcal{C}^{(k)}$ and let $f : \{0,1\}^{m(k)} \times D \times \{0,1\}$ be a universal hash function. Let $Y, Z \leftarrow U_{m(k)}, B \leftarrow U_m$, and $D_i \leftarrow \mathcal{C}_{h_i}^{(k)}$. Let $L(k) = \min_i H_\infty(D_i)$, and note that $L(k) = \omega(\log k)$. Then the Leftover Hash Lemma ([33]) implies that

$$\Delta(\langle Y, f_Y(D_1), ..., f_Y(D_m)\rangle, \langle Y, B\rangle) \leq m2^{-L(k)/2+1} ,$$

from which it is immediate that if we choose $Y \leftarrow U_{m(k)}$ once and publicly, then for all $1 \leq i \leq m$, f_Y will have negligible bias for $\mathcal{C}_{h_i}$ except with negligible probability. □

Lemma 4.8. If f is ϵ-biased on $\mathcal{C}_h$ for all h, then for any k and $s_1, s_2, \ldots, s_l$:

$$\Delta(\texttt{Basic\_Encode}(U_l, h, k), \mathcal{C}_h^l) \leq \epsilon l .$$

Proof. To see that this is so, imagine that the i^{th} bit of the input to `Basic_Encode`, c_i, was chosen so that $\Pr[c_i = 0] = \Pr[f(\mathcal{C}_{h_i}) = 0]$. In this case the the i^{th} document output by `Basic_Encode` will come from a distribution identical to $\mathcal{C}_{h_i}$. But since $\Delta(c_i, U_1) \leq \epsilon$, it must be the case that $\Delta(s_i, \mathcal{C}_{h_i}) \leq \epsilon$ as well, by proposition 2.4. The statistical distance between the entire sequences must then be at most ϵl, by the triangle inequality. □

Using these lemmata, we will show that public-key steganography is possible in any channel that is always informative. We note that procedure `Basic_Encode` has a small probability of failure: `Basic_Decode`(`Basic_Encode`(c, h, k)) might not equal c. This probability of failure, however, is negligible in k.

4.2.4 Chosen Hiddentext security

Let $E_{PK}(\cdot)$ and $D_{SK}(\cdot)$ denote the encryption and decryption algorithms for a public-key cryptosystem $\mathcal{E}$ which is indistinguishable from random bits under chosen plaintext attack (IND\$-CPA). Let ℓ be the expansion function of $\mathcal{E}$, i.e., $|E_{PK}(m)| = \ell(|m|)$. The following procedures allow encoding and decoding of messages in a manner which is steganographically secret under chosen hiddentext attack for the channel distribution $\mathcal{C}$:

Construction 4.9. (Chosen Hiddentext Security)

Procedure `CHA_Encode`:	**Procedure** `CHA_Decode`:
Input: $m \in \{0,1\}^*$, $h \in D^*$, key PK	**Input:** $s_1, \ldots, s_l \in D^l$, secret key SK
Let $c = E_{PK}(m)$	Let $c = \texttt{Basic\_Decode}(s_1, \ldots, s_l)$
Output: $\texttt{Basic\_Encode}(c, h, k)$	**Output:** $D_{SK}(c)$

Lemma 4.10. Assume f is ϵ-biased on $\mathcal{C}_h$ for all h. For any warden $W \in \mathcal{W}(t, l)$, we can construct an IND\$-CPA adversary A where

$$\mathbf{Adv}^{\mathsf{cha}}_{\mathsf{CHA},\mathcal{C},W}(k) \leq \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E},A}(k) + \ell(l)\epsilon.$$

The running time of A is at most $t + O(kl)$ (k is the security parameter of (PK, SK) and is used as a bound for `Basic_Encode`), and A makes q encryption queries for a total of l bits of plaintext.

Proof. We are given a program W for distinguishing between the output of Construction 4.9 and the ordinary traffic of the channel, and an oracle for sampling blocks from that channel. We construct a program A which plays the IND\$-CPA game: distinguishing $E_{PK}(m_W)$ oracle from $U_{\ell(l)}$. $A(PK)$ simply runs $W(PK)$, to get (m_W, h_W) and returns m_W for m_A. Then $A(PK, c)$ uses the oracle $\mathcal{C}_h$ to compute $s = \texttt{Basic\_Encode}(c, h_W, k)$, and returns the output of $W(PK, s)$. Consider the cases for A's input.

- If $c \leftarrow E_{PK}(m_W)$, then $s \leftarrow \texttt{CHA\_Encode}(PK, m_W, h_W)$; so we have

$$\Pr[A(PK, E_{PK}(m_A)) = 1] = \Pr[W(PK, SE(PK, m_W, h_W)) = 1] \ .$$

- If $c \leftarrow U_\ell$, then s is distributed identically to $\texttt{Basic\_Encode}(U_\ell, h_W, k)$; thus

$$\left|\Pr[A(PK, U_\ell) = 1] - \Pr[W(PK, C^\ell_{h_W}) = 1]\right| \leq \ell\epsilon$$

by lemma 4.8.

Combining the cases, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cha}}_{\mathtt{CHA},\mathcal{C},W}(k) &= |\Pr[W(PK, SE(PK, m_W, h_W)) = 1] - \Pr[W(PK, C^{\ell}_{h_W}) = 1]| \\
&\leq |\Pr[W(PK, SE(PK, m_W, h_W)) = 1] - \Pr[A(PK, U_\ell) = 1]| + \ell(l)\epsilon \\
&= |\Pr[A(PK, E_{PK}(m_A)) = 1] - \Pr[A(PK, U_\ell) = 1]| + \ell(l)\epsilon \\
&= \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) + \ell(l)\epsilon \ ,
\end{aligned}$$

as claimed. □

Theorem 4.11. *If f is ϵ-biased on $\mathcal{C}_h$ for all h, then*

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathtt{CHA},\mathcal{C}}(t, l, k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kl), l, k) + \ell(l)\epsilon \ .$$

4.3 Steganographic Key Exchange

In many cases in which steganography might be desirable, it may not be possible for either Alice or Bob to publish a public key without raising suspicion. In these cases, a natural alternative to public-key steganography is *steganographic key exchange*: Alice and Bob exchange a sequence of messages, indistinguishable from normal communication traffic, and at the end of this sequence they are able to compute a shared key. So long as this key is indistinguishable from a random key to the warden, Alice and Bob can proceed to use their shared key in a symmetric-key stegosystem. In this section, we will formalize this notion.

Definition 4.12. (Steganographic Key Exchange Protocol) A *steganographic key exchange protocol,* or SKEP $\mathcal{S}$, is a pair of efficient probabilistic algorithms:

- $\mathcal{S}$.`Encode_Key` (Abbreviated SE): takes as input a security parameter 1^k and a string of random bits. $SE(1^k, U_k)$ outputs a sequence of $l(k)$ documents.
- $\mathcal{S}$.`Compute_Key`: (Abbreviated SD): takes as input a security parameter 1^k, a string of random bits, and a sequence s of $l(k)$ documents. $SD(1^k, s, U_k)$ outputs an element of the key space $\mathcal{K}$.

We say that $\mathcal{S}$ is correct if these algorithms satisfy the property that there exists a negligible function $\mu(k)$ satisfying:

$$\Pr_{r_a,r_b}[SD(1^k, r_a, SE(1^k, r_b)) = SD(1^k, r_b, SE(1^k, r_a))] \geq 1 - \mu(k) \ .$$

We call the output of $SD(1^k, r_a, SE(1^k, r_b))$ the *result* of the protocol, and denote this result by $S_{KE}(r_a, r_b)$. We denote by $\mathcal{S}(1^k, r_a, r_b)$ the triple $(SE(1^k, r_a),\ SE(1^k, r_b),\ S_{KE}(r_a, r_b))$.

Alice and Bob perform a key exchange using $\mathcal{S}$ by sampling private randomness r_a, r_b, asynchronously sending $SE(1^k, r_a)$ and $SE(1^k, r_b)$ to each other, and using the result of the protocol as a key. Notice that in this definition a SKEP must be an asynchronous single-round scheme, ruling out multi-round key exchange protocols. This is for ease of exposition only.

We remark that many *authenticated* cryptographic key exchange protocols require three flows without a public-key infrastructure. Our SKE scheme will be secure with only two flows because we won't consider the same class of attackers as these protocols; in particular we will not worry about active attackers who alter the communications between Alice and Bob, and so Diffie-Hellman style two-flow protocols are possible. This may be a more plausible assumption in the SKE setting, since an attacker will not even be able to detect that a key exchange is taking place, while cryptographic key exchanges are typically easy to recognize.

Let W be a warden running in time t. We define W's *SKE advantage against* $\mathcal{S}$ on bidirectional channel $\mathcal{B}$ with security parameter k by:

$$\mathbf{Adv}^{\mathsf{ske}}_{\mathcal{S},\mathcal{B},W}(k) = \left| \Pr_{r_a,r_b}[W(\mathcal{S}(1^k, r_a, r_b)) = 1] - \Pr_K[W(\mathcal{B}, K) = 1] \right| \ .$$

We remark that, as in our other definitions, W also has access to bidirectional channel oracles $\mathcal{C}^a, \mathcal{C}^b$.

Let $\mathcal{W}(t)$ denote the set of all wardens running in time t. The *SKE insecurity of* $\mathcal{S}$ *on bidirectional channel* $\mathcal{B}$ *with security parameter* k is given by $\mathbf{InSec}^{\mathsf{ske}}_{\mathcal{S},\mathcal{B}}(t, k) = \max_{W \in \mathcal{W}(t)} \left\{ \mathbf{Adv}^{\mathsf{ske}}_{\mathcal{S},\mathcal{B},W}(k) \right\}$.

Definition 4.13. (Secure Steganographic Key Exchange) A SKEP $\mathcal{S}$ is said to be

(t, ϵ)*-secure for bidirectional channel* $\mathcal{B}$ if $\mathbf{InSec}^{\mathsf{ske}}_{\mathcal{S},\mathcal{B}}(t,k) \leq \epsilon(k)$. $\mathcal{S}$ is said to be secure for $\mathcal{B}$ if for all PPT adversaries W, $\mathbf{Adv}^{\mathsf{ske}}_{\mathcal{S},\mathcal{B},W}(k)$ is negligible in k.

4.3.1 Construction

The idea behind behind the construction for steganographic key exchange is simple: let g generate $\mathbb{Z}_P^*$, let Q be a large prime with $P = rQ + 1$ and r coprime to Q, and let $\hat{g} = g^r$ generate the subgroup of order Q. Alice picks random values $a \in \mathbb{Z}_{P-1}$ uniformly at random until she finds one such that $g^a \mod P$ has its most significant bit (MSB) set to 0 (so that $g^a \mod P$ is uniformly distributed in the set of bit strings of length $|P|-1$). She then uses `Basic_Encode` to send all the bits of $g^a \mod P$ except for the MSB (which is zero anyway). Bob does the same and sends all the bits of $g^b \mod P$ except the most significant one (which is zero anyway) using `Basic_Encode`. Bob and Alice then perform `Basic_Decode` and agree on the key value $\hat{g}^{ab}$:

Construction 4.14. (Steganographic Key Exchange)

Procedure `SKE_Encode`:	**Procedure** `SKE_Compute_Key`:
Input: primes P, Q, h, $g \in \mathbb{Z}_P^*$	**Input:** Stegotext $s_1, \ldots, s_l$; $a \in \mathbb{Z}_{P-1}$
repeat:	Let $c_b =$ `Basic_Decode`$(s_1, \ldots, s_l)$
sample $a \leftarrow U(\mathbb{Z}_{P-1})$	**Output:** $c_b^{ra} \mod P = \hat{g}^{ab}$
until MSB of $g^a \mod P$ equals 0	
Let $c_a =$ all bits of g^a except MSB	
Output: `Basic_Encode`(c_a, h, k)	

Lemma 4.15. Let f be ϵ-biased on $\mathcal{B}$. Then for any warden $W \in \mathcal{W}(t)$, we can construct a DDH adversary A where $\mathbf{Adv}^{\mathsf{ddh}}_A(\hat{g}, P, Q) \geq \frac{1}{4}\mathbf{Adv}^{\mathsf{ske}}_{\mathtt{SKE},\mathcal{B},W}(k) - 2k\epsilon$. The running time of A is at most $t + O(k^2)$.

Proof. A takes as input a triple $(\hat{g}^a, \hat{g}^b, \hat{g}^c)$ and attempts to decide whether $c = ab$, as follows. First, A computes $\hat{r}$ as the least integer such that $r\hat{r} = 1 \mod Q$, and then picks $\alpha, \beta \leftarrow \mathbb{Z}_r$. Then A computes $c_a = (\hat{g}^a)^{\hat{r}} g^{\alpha Q}$ and $c_b = (\hat{g}^b)^{\hat{r}} g^{\beta Q}$. If $c_a > 2^{k-1}$ or $c_b > 2^{k-1}$, A outputs 0. Otherwise, A computes $s_a =$ `Basic_Encode`(c_a), and $s_b =$ `Basic_Encode`(c_b); A then outputs the result of computing $W(s_a, s_b, \hat{g}^c)$. We claim that:

- The element c_a, c_b are uniformly chosen element of $\mathbb{Z}_P^*$, when $a, b \leftarrow \mathbb{Z}_Q$. To see that this is true, observe that the exponent of s_a, $\xi_a = r\hat{r}a + \alpha Q$, is congruent to $a \bmod Q$ and $\alpha Q \bmod r$; and that for uniform α, αQ is also a uniform residue mod r. By the Chinese remainder theorem, there is exactly one element of $\mathbb{Z}_{rQ} = \mathbb{Z}_{P-1}$ that satisfies these conditions, for every a and α. Thus c_a is uniformly chosen. The same argument holds for c_b.

- B halts and outputs 0 with probability at most $\frac{3}{4}$ over input and random choices; and conditioned on not halting, the values c_a, c_b are uniformly distributed in $\{0,1\}^k$. This is true because $2^k/P < \frac{1}{2}$, by assumption.

- The sequence (s_a, s_b) is $2k\epsilon$ statistically close to $\mathcal{B}$. This follows because of Lemma 4.8.

- When $c = ab$, the element $\hat{g}^c$ is exactly the output of $SD(a, s_b) = SD(b, s_a)$. This is because

$$\begin{aligned} c_a^{rb} &= (g^{r\hat{r}a+\alpha Q})^{rb} \\ &= g^{(\gamma Q+1)rab+rQ(\alpha b)} \\ &= g^{rab} = \hat{g}^c \end{aligned}$$

- When $c \neq ab$, the input $H\|s\|E_{H(\hat{g}^z)}(m_A))$ is selected exactly according to the output of H_1, by construction.

Thus,

$$\Pr[A(\hat{g}^a, \hat{g}^b, \hat{g}^{ab}) = 1] = \left(\frac{2^k}{P}\right)^2 \Pr[W(\mathcal{S}(a,b)) = 1] \ ,$$

and

$$\left|\Pr[A(\hat{g}^a, \hat{g}^b, \hat{g}^c) = 1] - \Pr_K[W(\mathcal{B}, K) = 1]\right| \leq 2k\epsilon \ .$$

And therefore $\mathbf{Adv}_A^{\mathsf{ddh}}(\hat{g}, P, Q) \geq \frac{1}{4}\mathbf{Adv}_{\mathcal{S},\mathcal{B},W}^{\mathsf{ske}}(k) - 2k\epsilon$. $\square$

Theorem 4.16. *If f is ϵ-biased on $\mathcal{B}$, then*

$$\mathbf{InSec}_{\mathsf{SKE},\mathcal{B}}^{\mathsf{ske}}(t,k) \leq 4\mathbf{InSec}_{\hat{g},P,Q}^{\mathsf{ddh}}(t + O(k^2))) + 8k\epsilon \ .$$

Chapter 5

Security against Active Adversaries

The results of the previous two chapters show that a *passive* adversary (one who simply eavesdrops on the communications between Alice and Bob) cannot hope to subvert the operation of a stegosystem. In this chapter, we consider the notion of an *active* adversary who is allowed to introduce new messages into the communications channel between Alice and Bob. In such a situation, an adversary could have two different goals: disruption or detection.

Disrupting adversaries attempt to prevent Alice and Bob from communicating steganographically, subject to some set of publicly-known restrictions. We call a stegosystem which is secure against this type of attack *robust*. In this chapter we will give a formal definition of *robustness* against such an attack, consider what type of restrictions on an adversary are *necessary* (under this definition) for the existence of a robust stegosystem, and give the first construction of a provably robust stegosystem against any set of restrictions satisfying this necessary condition. Our protocol is secure assuming the existence of pseudorandom functions.

Distinguishing adversaries introduce additional traffic between Alice and Bob in hopes of tricking them into revealing their use of steganography. We consider the security of symmetric- and public-key stegosystems against active distinguishers, and give constructions that are secure against various notions of active distinguishing attacks. We also show that *no stegosystem can be simultaneously secure against both disrupting and distinguishing active adversaries.*

5.1 Robust Steganography

Robust steganography can be thought of as a game between Alice and Ward in which Ward is allowed to make some alterations to Alice's messages. Ward wins if he can sometimes prevent Alice's hidden messages from being read; while Alice wins if she can pass a hidden message with high probability, even when Ward alters her public messages. For example, if Alice passes a single bit per document and Ward is unable to change the bit with probability at least $\frac{1}{2}$, Alice may be able to use error correcting codes to reliably transmit her message. It will be important to state the limitations we impose on Ward, since otherwise he can replace all messages with a new (independent) draw from the channel distribution, effectively destroying any hidden information. In this section we give a formal definition of robust steganography with respect to a limited adversary.

We will model the constraint on Ward's power by a relation R which is constrained to not corrupt the channel too much. That is, if Alice sends document d, Bob must receive a document d' such that $(d, d') \in R$. This general notion of constraint is sufficient to include many simpler notions such as (for example) "only alter at most 10% of the bits". We will assume that it would be feasible for Alice and Bob to check (after the fact) if in fact, Ward has obeyed this constraint; thus both Alice and Bob know the "rules" Ward must play by. Note however, that Ward's *strategy* is still unknown to Alice and Bob.

We consider robustness in a symmetric-key setting only, since unless Alice and Bob share some initial secret they cannot hope to accurately exchange keys. One could alternatively consider a scenario in which the adversary is not allowed to alter some initial amount of communications between Alice and Bob; but in this case, using a steganographic key exchange followed by a symmetric-key robust stegosystem is sufficient.

5.1.1 Definitions for Substitution-Robust Steganography

We model an R-bounded active warden W as an adversary which plays the following game against a stegosystem $\mathcal{S}$:

1. W is given oracle access to the channel distribution $\mathcal{C}$ and to $SE(K, \cdot, \cdot)$. W may access these oracles at any time throughout the game.

2. W presents an arbitrary message $m_W \in \{0,1\}^{l_2}$ and history h_W.

3. W is then given a sequence of documents $\sigma = (\sigma_1, \ldots \sigma_\ell) \leftarrow SE(K, m_W, h_W)$, and produces a sequence $s_W = (s_1, \ldots, s_\ell) \in D^\ell$, where $(\sigma_i, s_i) \in R$ for each $1 \leq i \leq \ell$.

Define the success of W against $\mathcal{S}$ by

$$\mathbf{Succ}^R_{\mathcal{S},W}(k) = \Pr[SD(K, s'_W, h_W) \neq m_W] ,$$

where the probability is taken over the choice of K and the random choices of $\mathcal{S}$ and W. Define the failure rate of S by

$$\mathbf{Fail}^R_{\mathcal{S}}(t, q, l, \mu, k) = \max_{W \in \mathcal{W}(R,t,q,l,\mu)} \left\{ \mathbf{Succ}^R_{\mathcal{S},W}(k) \right\} ,$$

where $\mathcal{W}(R, t, q, l)$ denotes the set of all R-bounded active wardens that submit at most $q(k)$ encoding queries of total length at most $l(k)$, produce a plaintext of length at most $\mu(k)$ and run in time at most $t(k)$.

Definition 5.1. A sequence of stegosystems $\{S_k\}_{k \in \mathbb{N}}$ is called *substitution robust* for $\mathcal{C}$ against R if it is steganographically secret for $\mathcal{C}$ and there is a negligible function $\nu(k)$ such that for every PPT W, for all sufficiently large k, $\mathbf{Succ}^R_{\mathcal{S},W}(k) < \nu(k)$.

5.1.2 Necessary conditions for robustness

Consider the question of what conditions on the relation R are necessary to allow communication to take place between Alice and Bob. Surely it should not be the case that $R = D \times D$, since in this case Ward's "substitutions" can be chosen independently of Alice's transmissions, and Bob will get no information about what Alice has said.

Furthermore, if there is some document d' and history h for which

$$\sum_{(d,d') \in R} \Pr_{\mathcal{C}_h}[d] = 1$$

then when h has transpired, Ward can effectively prevent the transfer of information from Alice to Bob by sending the document d' regardless of the document transmitted by Alice, because the probability Alice picks a document related to d' is 1. That is, after history h, regardless of Alice's transmission d, Ward can replace it by d', so seeing d' will give Bob no information about what Alice said.

Since we model the attacker as controlling the history h, then, a necessary condition on R and $\mathcal{C}$ for robust communication is that

$$\forall h. \Pr_{\mathcal{C}}[h] = 0 \text{ or } \max_y \sum_{(x,y)\in R} \Pr_{\mathcal{C}_h}[x] < 1 \ .$$

We denote by $\mathcal{I}(R, \mathcal{D})$ the function $\max_y \sum_{(x,y)\in R} \Pr_{\mathcal{D}}[x]$. We say that the pair $(R, \mathcal{D})$ is δ-admissible if $\mathcal{I}(R, \mathcal{D}) \leq \delta$ and a pair $(R, \mathcal{C})$ is δ-admissible if $\forall h \ \ \Pr_{\mathcal{C}}[h] = 0$ or $\mathcal{I}(R, \mathcal{C}_h) \leq \delta$. Our necessary condition states that $(R, \mathcal{C})$ must be δ-admissible for some $\delta < 1$.

It turns out that this condition (on R) will be sufficient, for an efficiently sampleable channel, for the existence of a stegosystem which is substitution-robust against R.

5.1.3 Universally Substitution-Robust Stegosystem

In this section we give a stegosystem which is substitution robust against any admissible bounding relation R, under a slightly modified assumption on the channel, and assuming that Alice and Bob know some efficiently computable, δ-admissible relation R' such that R' is a superset of R. As with most of our constructions, this stegosystem is not really practical but it serves as a proof that robust steganography is possible for any admissible relation.

Suppose that the channel distribution $\mathcal{C}$ is efficiently sampleable. (Recall that $\mathcal{C}$ is efficiently sampleable if there is an efficient algorithm $\mathtt{C}$ such that, given a uniformly chosen string $s \in \{0,1\}^k$, a security parameter 1^k and history h, $\mathtt{C}(h, 1^k, s)$ is indistinguishable from $\mathcal{C}_h$). We will assume that Alice, Bob, and Ward all have access to this algorithm. Furthermore, we assume Alice and Bob share a key K to a pseudorandom function family $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$; and have a synchronized counter N. We will let $n(k) = \omega(\log k)$ be a "robustness parameter." We begin with a stegosystem which robustly encodes a single bit.

Construction 5.2. `ROneBit`

Procedure `Encode`:	**Procedure** `Decode`:
Input: K, $m \in \{0,1\}$, h	**Input:** K, $s_1, \ldots, s_n \in D^n$, h
for $i = 1 \ldots n$ do	set $m = 0$ Set $h_0 = h_1 = ()$
set $\sigma_i = \mathtt{C}(h, 1^k, F_K(N, m))$	for $i = 1 \ldots n$ do
increment N	for $b \in \{0,1\}$ do
set $h = (h, s_i)$	set $\sigma_b = \mathtt{C}((h, h_b), 1^k, F_K(N, b))$
Output: $\sigma_1, \ldots, \sigma_n$	set $h_b = (h_b, \sigma_b)$
	if $(m = 0 \wedge (\sigma_0, s_i) \in R')$ then $m = 0$
	else $m = 1$
	set $h = (h, h_m)$
	Output: m

The idea behind this construction is this: suppose that instead of sharing a key to a pseudorandom function F, Alice and Bob shared two secret documents d_0, d_1 drawn independently from $\mathcal{C}_h$. Then Alice could send Bob the message bit m by sending document d_m, and Bob could recover m by checking to see if the document he received was related (by R') to d_0 or d_1. Since the adversary is R bounded and $(\mathcal{D}, R')$ is δ-admissible, the probability of a decoding error — caused either by the adversary, or by accidental draw of d_0, d_1 — would be at most δ. Intuitively, `ROneBit` reduces the probability of decoding error to δ^n by encoding each hiddentext bit n times.

Lemma 5.3. `ROneBit` is steganographically secret against a nonce-respecting chosen hiddentext attack:

$$\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{ROneBit},\mathcal{C}}(t, q, q) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(nq), nq).$$

Proof. Let W be a passive warden which runs in time t, and makes at most q queries of total length at most q (each query can be only one bit, because of the type). We construct a PRF adversary A which runs in time $t + O(nq)$ and makes at most nq queries to F, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) = \mathbf{Adv}^{\mathsf{ss}}_{S,\mathcal{C},W}(k)\ .$$

The PRF adversary takes a function oracle f, and emulates $W(1^k)$, responding to the queries W makes to the encoder SE by using f in place of $F_K(\cdot,\cdot)$. More formally, we define the subroutine $SSE^f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ as follows:

Procedure SSE^f:
Input: bit m, history h
for $i = 1 \ldots n$ do
  set $\sigma_i = \mathtt{C}(1^k, h, f(N, m))$
  increment N
  set $h = (h, \sigma_i)$
Output: $\sigma_1, \ldots, \sigma_n$

Then we define $A^f(1^k) = W^{SSE^f}(1^k)$; A's advantage over F is then:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \left|\Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\ &= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\ &= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| \\ &= \mathbf{Adv}^{\mathsf{ss}}_{\mathtt{ROneBit},\mathcal{C},W}(k) \ .\end{aligned}$$

Where the following cases for f justify the substitutions:

- f is chosen from $F_K(\cdot,\cdot)$. Then the output of SSE^f is distributed identically to the encoding function of `ROneBit`. That is,

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1] \ .$$

- f is chosen uniformly. Then by assumption on $\mathtt{C}$, the output of SSE^f is distributed identically to samples from $\mathcal{C}^n_h$. that is,

$$\Pr[A^f(1^k) = 1] = \Pr[W^{CT}(1^k) = 1] \ .$$

The claim follows by the definition of insecurity. □

Lemma 5.4. $\mathbf{Fail}^R_{\mathtt{ROneBit}}(t, q, q, 1, k) \leq \mathbf{InSec}^{\mathsf{prf}}_F(t + O(nq), nq, k) + \delta^n$.

Proof. Let W be an active R-bounded $(t, q, q, 1)$ warden. We construct a PRF adversary A which runs in time $t + O(nq)$, makes at most nq PRF queries, and satisfies $\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) \geq \mathbf{Succ}^R_{\mathtt{ROneBit},W}(k) - \delta^n$. A^f works by emulating W, using its function oracle f in place of $F_K(\cdot,\cdot)$ to emulate `ROneBit.Encode` in responding to the queries of W. Let m_W, s_W be the hiddentext and the stegotext sequence returned by W, respectively. Then A^f returns 1 iff $SD^f(s_W, h_W) \neq m_W$. Consider the following two cases for f:

- f is chosen uniformly from all appropriate functions. Then, for each i, the stegotexts $\sigma_i = \mathtt{C}(1^k, h_i, f(N+i, m_W))$ are distributed independently according to $\mathcal{C}_{h_i}$. Consider the sequence of "alternative stegotexts" $d_i = \mathtt{C}(1^k, h'_i, f(N + i, 1 - m_W))$; each of these is also distributed independently according to $\mathcal{C}_{h'_i}$; and since W is never given access to the d_i, the s_i are independent of the d_i. Now SD will fail (causing $A^f(1^k)$ to output 1) only if the event $\forall i.(d_i, s_i) \in R'$ occurs. Because the d_i are independent of the actions of W, and because $(\mathcal{D}, R')$ is δ-admissible, each event $(d_i, s_i) \in R'$ happens independently with probability at most δ. So the probability of failure is at most δ^n:

$$\Pr[A^f(1^k) = 1] \leq \delta^n \ .$$

- f is chosen uniformly from F_K. Then $A^F(1^k)$ outputs 1 exactly when W succeeds against `ROneBit`, by the definition of `ROneBit`.

$$\Pr[A^{F_K}(1^k) = 1] = \mathbf{Succ}^R_{\mathtt{ROneBit},W}(k) \ .$$

Taking the difference of these probabilities, we get:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1] \\
&= \mathbf{Succ}^R_{\mathtt{ROneBit},W}(k) - \Pr[A^f(1^k) = 1] \\
&\geq \mathbf{Succ}^R_{\mathtt{ROneBit},W}(l) - \delta^n \ .
\end{aligned}$$

□

Theorem 5.5. *If F is pseudorandom then* `ROneBit` *is substitution-robust against R for $\mathcal{C}$.*

Proof. The theorem follows by the conjunction of the previous lemmata. □

We now show how to extend ROneBit to handle multiple-bit messages. We assume the same setup as previously, i.e., Alice and Bob share a synchronized counter N and a key K to a PRF $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$; and know an efficiently computable, admissible relation $R' \supseteq R$. We assume that the "state updates" made by calls to ROneBit are maintained across invocations.

Construction 5.6. Robust

Procedure Encode:	Procedure Decode:
Input: K, $m \in \{0,1\}^l$, h	**Input:** K, $s_{1,1}, \ldots, s_{l,n} \in D^{l\times n}$, h
for $i = 1 \ldots l$ do	for $i = 1 \ldots l$ do
set $\sigma_{i,1\ldots n} = \texttt{ROneBit}.SE(K, m, h, N)$	set $m_i = \texttt{ROneBit}.SD(K, s_{i,1\ldots n}, h, N)$
Output: $\sigma_{1,1}, \ldots, \sigma_{l,n}$	**Output:** $m_1, \ldots, m_l$

Lemma 5.7. Robust is steganographically secret against a nonce-respecting chosen hiddentext attack:

$$\mathbf{InSec}^{\mathsf{ss}}_{\texttt{Robust},\mathcal{C}}(t,q,l,k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(nl), nl, k).$$

Proof. Suppose we are given a warden $W \in \mathcal{W}(t,q,l)$ against the stegosystem Robust. Then we can construct a warden $X \in \mathcal{W}(t,l,l)$ against ROneBit. X^M works by simulating W, responding to each oracle query m, h by computing $h_0 = h$, and $\sigma_{i,1\ldots n} = M(m_i, h_{i-1})$, $h_i = h, \sigma_{i,1\ldots n}$ for $1 \leq i \leq |m|$, and returning $\sigma_1, \ldots, \sigma_|m|$. Consider the cases for X's oracle M:

- If $M \leftarrow$ ROneBit.Encode, then X's responses are distributed identically to those of Robust.Encode. Thus
$$\Pr[X^{ST}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1]\ .$$
- if $M \leftarrow \mathcal{C}_h^n$, then the response of X to query m, h is distributed identically to $\mathcal{C}_h^{|m|\times n}$. Thus
$$\Pr[X^{CT}(1^k) = 1] = \Pr[W^{CT}(1^k) = 1]\ .$$

Combining the cases, we have

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ss}}_{\texttt{ROneBit},\mathcal{C},X}(k) &= \left|\Pr[X^{ST}(1^k) = 1] - \Pr[X^{CT}(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{ss}}_{\texttt{Robust},\mathcal{C},W}(k)
\end{aligned}$$

Combining the fact that X makes l queries to `ROneBit.Encode` and runs in time $t + O(l)$ with the result of lemma 5.3, we get

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathtt{Robust},\mathcal{C},W}(k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(nl), nl, k) \ .$$

$\square$

Lemma 5.8. $\mathbf{Fail}^{R}_{\mathtt{Robust}}(t, q, l, \mu, k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(nl), nl, k) + \mu\delta^n$.

Proof. Let W be an active R-bounded $(t, q, q, 1)$ warden. We construct a PRF adversary A which runs in time $t + O(nl)$, makes at most nl PRF queries, and satisfies $\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) \geq \mathbf{Succ}^{R}_{\mathtt{Robust},W}(k) - \mu\delta^n$. A^f works by emulating W, using its function oracle f in place of $F_K(\cdot, \cdot)$ to emulate `Robust` in responding to the queries of W. Let m_W, s_W be the hiddentext and the stegotext sequence returned by W, respectively. Then A^f returns 1 iff $SD^f(s_W, h_W) \neq m_W$. Consider the following two cases for f:

- f is chosen uniformly from all appropriate functions. Then, for each i, the stegotexts $\sigma_{i,j} = \mathtt{C}(1^k, h_{i,j}, f(N + (i-1)n + j, m_{W,i}))$ are distributed independently according to $\mathcal{C}^n_{h_i}$. Consider the sequence of "alternative stegotexts" $d_{i,j} = \mathtt{C}(1^k, h'_{i,j}, f(N + (i-1)n + j, 1 - m_{W,i}))$; each of these is also distributed independently according to $\mathcal{C}_{h'_{i,j}}$; and since W is never given access to the $d_{i,j}$, the $s_{i,j}$ are independent of the $d_{i,j}$. Now SD will fail (causing $A^f(1^k)$ to output 1) only if the event $\forall j.(d_{i,j}, s_{i,j}) \in R'$ occurs for some i. Because the $d_{i,j}$ are independent of the actions of W, and because $(\mathcal{D}, R')$ is δ-admissible, each event $(d_{i,j}, s_{i,j}) \in R'$ happens independently with probability at most δ. So the probability of failure for any i is at most δ^n. A union bound then gives us:

  $$\Pr[A^f(1^k) = 1] \leq \mu\delta^n \ .$$

- f is chosen uniformly from F_K. Then $A^F(1^k)$ outputs 1 exactly when W succeeds against `Robust`, by the definition of `Robust`.

  $$\Pr[A^{F_K}(1^k) = 1] = \mathbf{Succ}^{R}_{\mathtt{Robust},W}(k) \ .$$

Taking the difference of these probabilities, we get:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1] \\ &= \mathbf{Succ}^R_{\mathtt{Robust},W}(k) - \Pr[A^f(1^k) = 1] \\ &\geq \mathbf{Succ}^R_{\mathtt{Robust},W}(k) - \mu\delta^n \ .\end{aligned}$$

□

5.2 Active Distinguishing Attacks

In an active distinguishing attack, Ward is allowed to ask Bob to decode innocent-looking messages, in order to discover whether the messages sent by Alice are stegotexts or covertexts. In the real world, it might be that Ward knows that if Alice is using steganography, she will encode a certain set of instructions in her message to Bob. Ward could perhaps intercept that message and try to alter the instructions, and then send the altered message to Bob and see how he reacts. If Bob follows Ward's instructions, Ward can conclude that Alice's message must have contained steganography, and if Bob doesn't react, Alice's message was probably innocent.

5.2.1 Chosen-covertext attacks

A Chosen-covertext attack is essentially the same as a chosen-hiddentext attack except that the adversary is given access to a decoding oracle, with the restriction that he can't invoke the decoding oracle on the "challenge" sequence which is either a stegotext or a covertext. We will give a formal definition of this concept, and a construction for any efficiently sampleable channel, assuming the existence of a symmetric or public-key encryption scheme which is indistinguishable from random bits under chosen-ciphertext attack.

Symmetric chosen-covertext attacks

In order to construct a stegosystem which is secure against chosen-covertext attacks, we will first need to introduce the notion of a cryptosystem which is indistinguishable from random bits under chosen-ciphertext attack.

IND\$-CCA **Security**

Definition. Let $\mathcal{E}$ be a symmetric encryption scheme. We define a chosen-ciphertext attack against $\mathcal{E}$ as a game played by an oracle adversary A. A is given oracle access to D_K and an encryption oracle e which is either:

- E_K: an oracle that returns $E_K(m)$.
- \$: an oracle that returns a sample from $U_{|E_K(m)|}$.

A is restricted so that he may not query D_K on the result of any query to E_K. We define A's CCA advantage against $\mathcal{E}$ by

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{E},A}(k) = \left|\Pr[A^{E_K,D_K}(1^k) = 1] - \Pr[A^{\$,D_K}(1^k) = 1]\right| ,$$

where $K \leftarrow U_k$, and define the CCA insecurity of $\mathcal{E}$ by

$$\mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t, q_e, q_d, \mu_e, \mu_d, k) = \max_{A \in \mathcal{A}(t,q_e,q_d,\mu_e,\mu_d)} \left\{\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{E},A}(k)\right\} ,$$

where $\mathcal{A}(t, q_e, q_d, l^*, \mu_e, \mu_d)$ denotes the set of adversaries running in time t, that make q_e queries of μ_e bits to e, and q_d queries of μ_d bits to D_K.

Then $\mathcal{E}$ is $(t, q_e, q_d, \mu_e, \mu_d, k, \epsilon)$*-indistinguishable from random bits under chosen ciphertext attack* if $\mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t, q_e, q_d, \mu_e, \mu_d, k) \leq \epsilon$. $\mathcal{E}$ is called *indistinguishable from random bits under chosen ciphertext attack* (IND\$-CCA) if for every PPT A, $\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k)$ is negligible in k.

Construction. We let $\mathcal{E}$ be any IND\$-CPA-secure symmetric encryption scheme and let $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ be a pseudorandom function. We let $K, \kappa \leftarrow U_k$. We construct a cryptosystem E as follows:

- $\mathsf{E}.\mathtt{Encrypt}(K, \kappa, m)$: Draw $r \leftarrow U_k$, $c \leftarrow \mathcal{E}.E(K, m)$, compute $t = F_\kappa(r\|c)$, and return $r\|c\|t$.
- $\mathsf{E}.\mathtt{Decrypt}(K, \kappa, r\|c\|t)$: If $F_\kappa(r\|c) = t$, then return $\mathcal{E}.D_K(c)$, else return $\perp$.

Theorem 5.9.

$$\mathbf{InSec}^{\mathsf{cca}}_{\mathsf{E}}(t,\vec{q},\vec{\mu},l^*,k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t',q_e,\mu_e,k) + 2\mathbf{InSec}^{\mathsf{prf}}_{F}(t',q_e+q_d,k) + (q_e^2+q_d)2^{-k}$$

Proof. Choose an arbitrary adversary $A \in \mathcal{A}(t,q_e,q_d,\mu_e,\mu_d)$. We will consider the advantage of A in distinguishing the following set of hybrid oracle pairs:

- $\mathsf{E}_1,\mathsf{D}_1$: $\mathsf{E}_1(m) = \mathsf{E}.\mathtt{Encrypt}(m)$, $\mathsf{D}_1(c) = \mathsf{E}.\mathtt{Decrypt}(c)$.
- $\mathsf{E}_2,\mathsf{D}_2$: uniformly choose $f : \{0,1\}^* \to \{0,1\}^k$, and a $K \leftarrow U_k$.

  To draw from $\mathsf{E}_2(m)$, choose $r \leftarrow U_k$, draw $c \leftarrow \mathcal{E}.E_K(m)$, compute $t = f(r\|c)$, and output $r\|c\|t$

  To compute $\mathsf{D}_2(r\|c\|t)$, output $\perp$ if $t \neq f(r\|c)$ and return $\mathcal{E}.D_K(c)$ otherwise.
- $\mathsf{E}_3,\mathsf{D}_3$: choose a random $f : \{0,1\}^* \to \{0,1\}^k$, and a random $K \leftarrow U_k$.

  To draw from $\mathsf{E}_3(m)$, choose $r \leftarrow U_k$, draw $c \leftarrow U_{\ell(|m|)}$, compute $t = f(r\|c)$, and output $r\|c\|t$.

  To compute $\mathsf{D}_3(m)$, output $\perp$ if $t \neq f(r\|c)$ and return $\mathcal{E}.D_K(c)$ otherwise.
- $\mathsf{E}_4,\mathsf{D}_4$: uniformly choose $f : \{0,1\}^* \to \{0,1\}^k$ and $K \leftarrow U_k$.

  To draw from $\mathsf{E}_4(m)$, choose $c \leftarrow U_{2k+\ell(|m|)}$.

  To compute $\mathsf{D}_4(r\|c\|t)$, output $\perp$ if $t \neq f(r\|c)$ and return $\mathcal{E}.D_K(c)$ otherwise.
- $\mathsf{E}_5,\mathsf{D}_5$: choose $K,\kappa \leftarrow U_k$.

  To draw from $\mathsf{E}_5(m)$, choose $c \leftarrow U_{2k+\ell(|m|)}$.

  To compute $\mathsf{D}_5(r\|c\|t)$, output $\perp$ if $t \neq F_\kappa(r\|c)$ and return $\mathcal{E}.D_K(c)$ otherwise.

By construction it is clear that

$$\Pr[A^{\mathsf{E}.E_K,\mathsf{E}.D_K}(1^k)=1] = \Pr[A^{\mathsf{E}_1,\mathsf{D}_1}(1^k)=1] \ ,$$

and it is also obvious that

$$\Pr[A^{\$,\mathsf{E}.D_K}(1^k)=1] = \Pr[A^{\mathsf{E}_5,\mathsf{D}_5}(1^k)=1] \ .$$

If we define the function

$$\mathbf{Adv}^{i}_{A}(k) = \left|\Pr[A^{\mathsf{E}_i,\mathsf{D}_i}(1^k) = 1] - \Pr[A^{\mathsf{E}_{i+1},\mathsf{D}_{i+1}}(1^k) = 1]\right| ,$$

we then have that:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cca}}_{A,\mathsf{E}}(k) &= \left|\Pr[A^{\mathsf{E}.E_K,\mathsf{E}.D_K}(1^k) = 1] - \Pr[A^{\$,\mathsf{E}.D_K}(1^k) = 1]\right| \\
&= \left|\Pr[A^{\mathsf{E}_1,\mathsf{D}_1}(1^k) = 1] - \Pr[A^{\mathsf{E}_5,\mathsf{D}_5}(1^k) = 1]\right| \\
&\le \sum_{i=1}^{4} \left|\Pr[A^{\mathsf{E}_i,\mathsf{D}_i}(1^k) = 1] - \Pr[A^{\mathsf{E}_{i+1},\mathsf{D}_{i+1}}(1^k) = 1]\right| \\
&= \sum_{i=1}^{4} \mathbf{Adv}^{i}_{A}(k)
\end{aligned}$$

We will proceed to bound $\mathbf{Adv}^{i}_{A}(k)$, for $i \in \{1, 2, 3, 4\}$.

Lemma 5.10. $\mathbf{Adv}^{1}_{A}(k) \le \mathbf{InSec}^{\mathsf{prf}}_{F}(t', q_e + q_d, k)$

Proof. We design a PRF adversary B such that $\mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) \ge \mathbf{Adv}^{1}_{A}(k)$ as follows. B picks $K \leftarrow U_k$ and runs A. B uses its function oracle f to respond to A's queries as follows:

- On encryption query m, B picks $r \leftarrow U_k$, computes $c \leftarrow \mathcal{E}.E_K(m)$, computes $t = f(r\|c)$ and returns $r\|c\|t$.
- On decryption query $r\|c\|t$, B returns $\perp$ if $t \ne f(r\|c)$ and returns $\mathcal{E}.D_K(c)$ otherwise.

Clearly, when B's oracle $f \leftarrow F$, B simulates $\mathsf{E}_1, \mathsf{D}_1$ to A:

$$\Pr[B^{F_K}(1^k) = 1] = \Pr[A^{\mathsf{E}_1,\mathsf{D}_1}(1^k) = 1] ,$$

and when $f \leftarrow U(*, k)$, B simulates $\mathsf{E}_2, \mathsf{D}_2$ to A:

$$\Pr[B^{f}(1^k) = 1] = \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1] ,$$

which gives us

$$\begin{aligned}
\mathbf{Adv}^{1}_{A}(k) &= \left|\Pr[A^{\mathsf{E}_1,\mathsf{D}_1}(1^k) = 1] - \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1]\right| \\
&= \left|\Pr[B^{F_K}(1^k) = 1] - \Pr[B^{f}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) \le \mathbf{InSec}^{\mathsf{prf}}_{F}(t', q_e + q_d, k)
\end{aligned}$$

as claimed. $\square$

Lemma 5.11. $\mathbf{Adv}^2_A(k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t', q_e, \mu_e, k) + q_d 2^{-k}$

Proof. We will construct a CPA adversary B for $\mathcal{E}$ such that

$$\mathbf{Adv}^{\mathsf{cpa}}_{B,\mathcal{E}}(k) \geq \mathbf{Adv}^2_A(k) - q_d 2^{-k} .$$

$B^{\mathcal{O}}$ works by emulating A, responding to queries as follows, where f is a randomly-chosen function built up on a per-query basis by B:

- on encryption query m, B picks $r \leftarrow U_k$, computes $c = \mathcal{O}(m)$, and sets $t = f(r\|c)$, and returns $r\|c\|t$.
- on decryption query $r\|c\|t$, B checks whether $t = f(c\|r)$; if not, B returns $\perp$ and otherwise B halts and outputs 0.

Let V denote the event that A submits a decryption query that would cause B to halt. Then, conditioned on $\neg\mathsf{V}$, when B's oracle is \$, B perfectly simulates $\mathsf{E}_3, \mathsf{D}_3$ to A:

$$\Pr[B^{\$}(1^k) = 1] = \Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1 | \neg\mathsf{V}] .$$

Also, conditioned on $\neg\mathsf{V}$, when B's oracle is $\mathcal{E}.E_K$, B perfectly simulates $\mathsf{E}_2, \mathsf{D}_2$ to A:

$$\Pr[B^{E_K}(1^k) = 1] = \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1 | \neg\mathsf{V}] .$$

Combining the cases, we have:

$$\begin{aligned}
\mathbf{Adv}^2_A(k) &= \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1] - \Pr[A^{\mathsf{E}_2,\mathsf{E}_2}(1^k) = 1]\right| \\
&= \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\mathsf{V}] \Pr[\mathsf{V}] + \Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\neg\mathsf{V}] \Pr[\neg\mathsf{V}] \right. \\
&\quad \left. - \left(\Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\mathsf{V}] \Pr[\mathsf{V}] + \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\neg\mathsf{V}] \Pr[\neg\mathsf{V}]\right)\right| \\
&\leq \Pr[\mathsf{V}] \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\mathsf{V}] - \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\mathsf{V}]\right| \\
&\quad + \Pr[\neg\mathsf{V}] \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\neg\mathsf{V}] - \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\neg\mathsf{V}]\right| \\
&\leq \Pr[\mathsf{V}] + \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\neg\mathsf{V}] - \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\neg\mathsf{V}]\right| \\
&\leq \Pr[\mathsf{V}] + \left|\Pr[B^{\$}(1^k) = 1] - \Pr[B^{E_K}(1^k) = 1]\right| \\
&\leq \Pr[\mathsf{V}] + \mathbf{Adv}^{\mathsf{cpa}}_{B,\mathcal{E}}(k) \\
&\leq q_d 2^{-k} + \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t', q_e, \mu_e, k)
\end{aligned}$$

Where the last line follows because each decryption query causes B to halt with probability 2^{-k}; the union bound gives the result. □

Lemma 5.12. $\mathbf{Adv}^3_A(k) \leq \frac{q_e^2}{2^k}$

Proof. Notice that unless E_3 chooses the same values of (r, c) at least twice, E_3 and E_4 are identical. Denote this event by C. Then we have:

$$\begin{aligned}
\mathbf{Adv}^3_A(k) &= \left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1] - \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1]\right| \\
&= \Big|\big(\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\mathsf{C}]\Pr[\mathsf{C}] + \Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\neg\mathsf{C}]\Pr[\neg\mathsf{C}]\big) \\
&\quad - \big(\Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1|\mathsf{C}]\Pr[\mathsf{C}] + \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1|\neg\mathsf{C}]\Pr[\neg\mathsf{C}]\big)\Big| \\
&\leq \Pr[\mathsf{C}]\left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\mathsf{C}] - \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1|\mathsf{C}]\right| \\
&\quad + \Pr[\neg\mathsf{C}]\left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\neg\mathsf{C}] - \Pr[A^{\mathsf{E}_2,\mathsf{D}_2}(1^k) = 1|\neg\mathsf{C}]\right| \\
&= \Pr[\mathsf{C}]\left|\Pr[A^{\mathsf{E}_3,\mathsf{D}_3}(1^k) = 1|\mathsf{C}] - \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1|\mathsf{C}]\right| \\
&\leq \Pr[\mathsf{C}] \\
&\leq 2^{-k}\binom{q_e}{2}
\end{aligned}$$

□

Lemma 5.13. $\mathbf{Adv}^4_A(k) \leq \mathbf{InSec}^{\mathsf{prf}}_F(t', q_d, k)$

Proof. We construct a PRF adversary B against F with advantage

$$\mathbf{Adv}^{\mathsf{prf}}_{B,F}(k) = \mathbf{Adv}^4_A(k)\ .$$

B^f starts by choosing $K \leftarrow U_k$. B then runs A, responding to encryption queries $E(m)$ with $r\|c\|t \leftarrow U_{2k+\ell(|m|)}$, and responding to decryption queries $D(r\|c\|t)$ with $\perp$ if $t \neq f(r\|c)$, and $D_K(c)$ otherwise. B outputs the bit chosen by A. Notice that by construction,

$$\Pr[B^{F_K}(1^k) = 1] = \Pr[A^{\mathsf{E}_5,\mathsf{D}_5}(1^k) = 1]\ \text{, and}$$
$$\Pr[B^f(1^k) = 1] = \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k) = 1]\ ,$$

so by definition of advantage, we get:

$$\begin{aligned}\mathbf{Adv}_A^4(k) &= \left|\Pr[A^{\mathsf{E}_5,\mathsf{D}_5}(1^k)=1] - \Pr[A^{\mathsf{E}_4,\mathsf{D}_4}(1^k)=1]\right| \\ &= \left|\Pr[B^{F_K}(1^k)=1] - \Pr[B^f(1^k)=1]\right| \\ &= \mathbf{Adv}_{B,F}^{\mathsf{prf}}(k) \le \mathbf{InSec}_F^{\mathsf{prf}}(t', q_d, k)\end{aligned}$$

□

The theorem follows by the conjunction of the lemmata. □

Chosen-covertext attack definition

In an adaptive chosen-covertext attack against a symmetric stegosystem $\mathcal{S}$, an adversary W is given access to a mystery oracle $\mathcal{O}$, which is either SE_K for a uniformly chosen key K or $\mathcal{O}_\mathcal{C}$, which on query m, h returns a sample from $\mathcal{C}_h^{|SE_K(m,h)|}$. The attacker is restricted to querying SD only on strings which were not generated by queries to $\mathcal{O}$. (As always, W is allowed to know the channel distribution $\mathcal{C}$) At the conclusion of the attack, W must guess the type of $\mathcal{O}$. We define the *Symmetric Chosen-Covertext Advantage* of W against $\mathcal{S}$ with respect to $\mathcal{C}$ by

$$\mathbf{Adv}_{\mathcal{S},\mathcal{C},\mathcal{W}}^{\mathsf{scca}}(k) = \left|\Pr[W^{SE,SD}(1^k)=1] - \Pr[W^{\mathcal{O}_\mathcal{C},SD}(1^k)=1]\right| \ ,$$

And define the sCCA insecurity of $\mathcal{S}$ with respect to $\mathcal{C}$ by

$$\mathbf{InSec}_{\mathcal{S},\mathcal{C}}^{\mathsf{scca}}(t, q_e, q_d, \mu_e, \mu_d, k) = \max_{W \in \mathcal{W}(t,\vec{q},\vec{\mu})} \left\{\mathbf{Adv}_{\mathcal{S},\mathcal{C},\mathcal{W}}^{\mathsf{scca}}(k)\right\} \ ,$$

where $\mathcal{W}(t,\vec{q},\vec{\mu})$ denotes the class of all W running in time t which make at most q_e encoding queries of μ_e bits and at most q_d decoding queries of μ_d bits.

We say that $\mathcal{S}$ is $(t,\vec{q},\vec{\mu},k,\epsilon)$ secure against symmetric chosen-covertext attack with respect to $\mathcal{C}$ if

$$\mathbf{InSec}_{\mathcal{S},\mathcal{C}}^{\mathsf{scca}}(t,\vec{q},\vec{\mu},k) \le \epsilon \ ,$$

and that $\mathcal{S}$ is secure against symmetric chosen-covertext attack with respect to $\mathcal{C}$ ($\mathsf{SS\text{-}sCCA}$) if $\mathbf{Adv}_{\mathcal{S},\mathcal{C},\mathcal{W}}^{\mathsf{scca}}(k)$ is negligible for all PPT W.

Construction

Both of our constructions of chosen-covertext secure stegosystems will require an efficiently sampleable channel $\mathcal{C}$,[1] and will make use of the following "deterministic encoding" routine:

Construction 5.14. (Deterministic Encoding)

> **Procedure** `DEncode`:
> **Input:** bits $c_1, \ldots, c_l$, history h, bound k, randomness $r_1, \ldots r_{lk} \in \{0,1\}^k$
> Let $\iota = 0$
> for $i = 1 \ldots l$ do
>   Let $j = 0$
>   repeat:
>   compute $s_i = \mathtt{C}(1^k, h, r_\iota)$, increment j, ι
>   until $f(s_i) = c_i$ OR $(j > k)$
>   set $h = h, s_i$
> **Output:** $s_1, s_2, \ldots, s_l$

For a given sequence of lk random inputs, this routine has exactly one possible encoding for any message. Building on this routine, we will assume that Alice and Bob share a key K for a symmetric IND\$-CCA-secure encryption scheme $\mathcal{E} = (G, E, D)$ such that for any key K, for any $l \geq k$, $\Pr[U_{\ell(l)} \in [E_K(\{0,1\}^l)]] \leq 1/2^k$ (An example is the scheme of the previous section). We will also assume that $G : \{0,1\}^k \to \{0,1\}^{k \times lk}$ is a pseudorandom generator. The the following scheme is secure against symmetric chosen-covertext attack:

Construction 5.15. (`sCCA` Stegosystem)

Procedure `Encode`:	**Procedure** `Decode`:
Input: $m \in \{0,1\}^\ell$, h, K	**Input:** $s_1, \ldots, s_l$, h, K
Choose $r \leftarrow U_k$	Let $c = \mathtt{Basic\_Decode}(s_1, \ldots, s_l)$
Let $c = E_K(r \| m)$	Parse $D_K(c)$ as $r \|_k m$.
Let $\vec{r} = G(r)$	Set $\vec{r} = G(r)$.
Output: $\mathtt{DEncode}(c, h, k, \vec{r})$	If $s \neq \mathtt{DEncode}(c, h, k, \vec{r})$ return $\perp$.
	Output: m

Theorem 5.16. *if f is ϵ-biased for $\mathcal{C}$, then*

$$\mathbf{InSec}^{\mathsf{scca}}_{\mathtt{sCCA},\mathcal{C}}(t, \vec{q}, \vec{\mu}, k) \leq \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t', \vec{q}, \vec{\mu}, k) + q_e 2^{-k} + \ell(\mu_e)\epsilon + q_e \mathbf{InSec}^{\mathsf{prg}}_{G}(t', k) \ ,$$

[1] Recall that a channel is efficiently sampleable if there exists a PPT $\mathtt{C}$ such that $\mathtt{C}(1^k, h, U_k)$ and $\mathcal{C}_h$ are computationally indistinguishable

where $t' \leq t + O((\mu_e + \mu_d)k)$.

Proof. Choose an arbitrary $W \in \mathcal{W}(t, \vec{q}, \vec{\mu})$. We will bound $\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},W}(k)$ by considering the following sequence of hybrid oracles:

- $\mathcal{O}_1$: the channel oracle $\mathcal{C}$.
- $\mathcal{O}_2$: responds to queries by replacing $c \leftarrow E_K(r\|m)$ with $c \leftarrow U$ and replacing $\vec{r} = G(r)$ with $\vec{r} \leftarrow U_{k \times lk}$.
- $\mathcal{O}_3$: responds to queries by replacing $c \leftarrow E_K(r\|m)$ with $c \leftarrow U$.
- $\mathcal{O}_4$: responds to queries with `sCCA.Encode`.

Clearly $\Pr[W^{\mathcal{O}_1, SD_K}(1^k) = 1] = \Pr[W^{\mathcal{O}_C, SD_K}(1^k) = 1]$ and $\Pr[W^{\mathcal{O}_4, SD_K}(1^k) = 1] = \Pr[W^{SE_K, SD_K}(1^k) = 1]$. Thus

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},W}(k) &= \left|\Pr[W^{\mathcal{O}_4, SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_1, SD_K}(1^k) = 1]\right| \\
&\leq \left|\Pr[W^{\mathcal{O}_2, SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_1, SD_K}(1^k) = 1]\right| \\
&\quad + \left|\Pr[W^{\mathcal{O}_3, SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_2, SD_K}(1^k) = 1]\right| \\
&\quad + \left|\Pr[W^{\mathcal{O}_4, SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_3, SD_K}(1^k) = 1]\right|
\end{aligned}$$

For convenience, we will define the quantity

$$\mathbf{Adv}^{i}_{W}(k) = \left|\Pr[W^{\mathcal{O}_{i+1}, SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_i, SD_K}(1^k) = 1]\right| \ ,$$

and we will proceed to bound $\mathbf{Adv}^{i}_{W}(k)$ for $i \in \{1, 2, 3\}$.

Lemma 5.17. $\mathbf{Adv}^{1}_{W}(k) \leq \ell(\mu_e)\epsilon$

Proof. This follows from lemma 4.8. □

Lemma 5.18. $\mathbf{Adv}^{2}_{W}(k) \leq q_e \mathbf{InSec}^{\mathsf{prg}}_{G}(t', k)$

Proof. We will construct a PRG adversary A for G such that

$$\mathbf{Adv}^{\mathsf{prg}}_{G,A}(k) \geq 1/q_e \mathbf{Adv}^{2}_{W}(k) \ .$$

A works as follows: first, A picks a key $K \leftarrow U_k$ to use in responding to the queries W makes to SD_K. Suppose A is given as input q_e strings $r_1, \ldots, r_{q_e}$ of length $k \times lk$ and asked to decide whether they are all samples from $U_{k\times lk}$ or samples from $G(U_k)$. Then A can achieve advantage precisely $\mathbf{Adv}^2_W(k)$ by emulating W, responding to its decoding queries using K, and responding to the i^{th} encoding query (m, h) by drawing $c \leftarrow U_{\ell(|m|+k)}$ and giving the response $\texttt{DEncode}(c, h, k, r_i)$. If all of the r_i are drawn from $U_{k\times lk}$, then A perfectly simulates $\mathcal{O}_1$ to W, and if all are drawn from $G(U_k)$, A perfectly simulates $\mathcal{O}_2$. Thus A's advantage in distinguishing $G(U_k)^{q_e}$ and $U^{q_e}_{k\times lk}$ is exactly $\mathbf{Adv}^2_W(k)$. The lemma follows from this fact and proposition 2.6 (a straightforward hybrid argument). $\square$

Lemma 5.19. $\mathbf{Adv}^3_W(k) \leq \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t', \vec{q}, \vec{\mu}, k) + q_e 2^{-k}$

Proof. We will construct an adversary A that plays the chosen-ciphertext attack game against $\mathcal{E}$ with advantage

$$\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k) \geq \mathbf{Adv}^3_W(k) \ .$$

A works by emulating W and responding to queries as follows:

- on encoding query (m, h), $A^{\mathcal{O}}$ chooses $r \leftarrow U_k$, computes $c \leftarrow \mathcal{O}(r\|m)$, and returns $\texttt{DEncode}(c, h, k, G(r))$.

- on decoding query (s, h), A computes $c = \texttt{Basic\_Decode}(s, h)$; if c was previously generated by an encoding query, A returns $\perp$, otherwise A uses its decryption oracle to compute $r\|_k m = D_K(c)$. If $c \neq \perp$ and $s = \texttt{DEncode}(c, h, k, G(r))$, A returns m, otherwise A returns $\perp$.

In other words, A simulates running the routines `sCCA.Encode` and `sCCA.Decode` with its oracles; with the exception that because A is playing the IND\$-CCA game, he is not allowed to query D_K on the result of an encryption query: thus a decoding query that has the same underlying ciphertext c must be dealt with specially.

Notice that when A is given an encryption oracle, he perfectly simulates $\mathcal{O}_4$ to W, that is:

$$\Pr[A^{E_K, D_K}(1^k) = 1] = \Pr[W^{\mathcal{O}_4, SD_K}(1^k) = 1] \ .$$

This is because when $c = E_K(r\|m)$ then the test $s = \texttt{DEncode}(c, h, k, G(r))$ would fail anyways.

Likewise, when A is given a random-string oracle, he perfectly simulates $\mathcal{O}_3$ to W, *given that the outputs of $\mathcal{O}$ are not valid ciphertexts.* Let us denote the event that some output of $\mathcal{O}$ is a valid ciphertext by V, and the event that some output of $\mathcal{O}_3$ encodes a valid ciphertext by U; notice that by construction $\Pr[\mathsf{U}] = \Pr[\mathsf{V}]$. We then have that

$$\begin{aligned}
\Pr[A^{\$,D_K}(1^k) = 1] &= \Pr[A^{\$,D_K}(1^k) = 1|\neg\mathsf{V}]\Pr[\neg\mathsf{V}] + \Pr[A^{\$,D_K}(1^k) = 1|\mathsf{V}]\Pr[\mathsf{V}] \\
&\leq \Pr[W^{\mathcal{O}_3,SD_K}(1^k) = 1|\neg\mathsf{U}]\Pr[\neg\mathsf{U}] + \Pr[\mathsf{V}] \\
&\leq \Pr[W^{\mathcal{O}_3,SD_K}(1^k) = 1] + \Pr[\mathsf{V}] \\
&\leq \Pr[W^{\mathcal{O}_3,SD_K}(1^k) = 1] + q_e 2^{-k} \ ,
\end{aligned}$$

since $\Pr[\mathsf{V}] \leq q_e 2^{-k}$ by assumption on $\mathcal{E}$ and the union bound.

Combining the cases, we find that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k) &= \Pr[A^{E_K,D_K}(1^k) = 1] - \Pr[A^{\$,D_K}(1^k) = 1] \\
&= \Pr[W^{\mathcal{O}_4,SD_K}(1^k) = 1] - \Pr[A^{\$,D_K}(1^k) = 1] \\
&\geq \Pr[W^{\mathcal{O}_4,SD_K}(1^k) = 1] - \Pr[W^{\mathcal{O}_3,SD_K}(1^k) = 1] - q_e 2^{-k} \\
&= \mathbf{Adv}^{3}_{W}(k) - q_e 2^{-k}
\end{aligned}$$

Which proves the lemma. □

Combining the three lemmata yields the proof of the theorem. □

Public-Key Chosen-covertext attacks

In the public-key case, we will likewise need to construct a public-key encryption scheme which is indistinguishable from random bits under chosen-ciphertext attack. The definitions in this section are mostly analogous to those of the previous section, although the construction of a public-key encryption scheme satisfying this definition uses very different techniques.

IND\$-CCA

Let $\mathcal{E}$ be a public-key encryption scheme. A chosen-ciphertext attack against $\mathcal{E}$ is defined analogously to the symmetric case, except that instead of an oracle for E_{PK}, the adversary A is given the public key PK: Let $\mathcal{E}$ be a symmetric encryption scheme. We define a chosen-ciphertext attack against $\mathcal{E}$ as a game played by an oracle adversary A:

1. A is given PK and oracle access to D_{SK}, and determines a *challenge message* m^* of length l^*.

2. A is given a *challenge ciphertext* c^*, which is either drawn from $E_{PK}(m^*)$ or $U_{\ell(l^*)}$.

3. A continues to query D_{SK} subject to the restriction that A may not query $D_{SK}(c^*)$. A outputs a bit.

We define A's CCA advantage against $\mathcal{E}$ by

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{E},A}(k) = \left|\Pr[A^{D_{SK}}(PK, E_{PK}(m^*)) = 1] - \Pr[A^{D_{SK}}(PK, U_\ell) = 1]\right| \ ,$$

where $m^* \leftarrow A^{D_{SK}}(PK)$ and $(PK, SK) \leftarrow G(1^k)$, and define the CCA insecurity of $\mathcal{E}$ by

$$\mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t, q, \mu, l^*, k) = \max_{A \in \mathcal{A}(t,q,,\mu,l^*)} \left\{\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{E},A}(k)\right\} \ ,$$

where $\mathcal{A}(t, q, \mu, l^*)$ denotes the set of adversaries running in time t, that make q queries of total length μ, and issue a challenge message m^* of length l^*. Then $\mathcal{E}$ is $(t, q, \mu, l^*, k, \epsilon)$-*indistinguishable from random bits under chosen ciphertext attack* if $\mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t, q, \mu, l^*, k) \le \epsilon$. $\mathcal{E}$ is called *indistinguishable from random bits under chosen ciphertext attack* (IND\$-CCA) if for every PPTM A, $\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k)$ is negligible in k.

Construction. Let Π_k be a family of trapdoor one-way permutations on domain $\{0,1\}^k$. Let $\mathcal{SE}_{k'} = (E, D)$ be a symmetric encryption scheme which is IND\$-CCA secure. Let $H : \{0,1\}^k \leftarrow \{0,1\}^{k'}$ be a random oracle. We define our encryption scheme $\mathcal{E}$ as follows:

- `Generate`(1^k): draws $(\pi, \pi^{-1}) \leftarrow \Pi_k$; the public key is π and the private key is π^{-1}.

- $\texttt{Encrypt}(\pi, m)$: draws a random $x \leftarrow U_k$, computes $K = H(x)$, $c = E_K(m)$, $y = \pi(x)$ and returns $y\|c$.
- $\texttt{Decrypt}(\pi^{-1}, y\|c)$: computes $x = \pi^{-1}(y)$, sets $K = H(x)$ and returns $D_K(c)$.

Theorem 5.20.

$$\mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t, q, \mu, l, k) \leq \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t, k) + \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{SE}}(t', 1, q, l, \mu, k) \ ,$$

where $t' \leq t + O(q_H)$.

Proof. We will show how to use any adversary $A \in \mathcal{A}(t, q, \mu, l)$ against $\mathcal{E}$ to create an adversary B which plays both the $\mathsf{IND\$\text{-}CCA}$ game against $\mathcal{SE}$ and the OWP game against Π so that B succeeds in at least one game with success close to that of A. B receives as input an element $\pi \in \Pi$ and a $y^* \in \{0,1\}^k$ and also has access to encryption and decryption oracles $\mathcal{O}, D_K$ for $\mathcal{SE}$. B keeps a list L of (y, z) pairs, where $y \in \{0,1\}^k$ and $z \in \{0,1\}^{k'}$, initially, L is empty. B runs A with input π and answers the decryption and random oracle queries of A as follows:

- When A queries $H(x)$, B first computes $y = pi(x)$, and checks to see whether $y^* = y$; if it does, B "decides" to play the OWP game and outputs x, the inverse of y^*. Otherwise, B checks to see if there is an entry in L of the form (y, z); if there is, B returns z to A. If there is no such entry, B picks a $z \leftarrow U_{k'}$, adds (y, z) to L and returns z to A.
- When A queries $D_{SK}(y\|c)$, first check whether $y = y^*$; if so, return $D_K(c)$. Otherwise, check whether there is an entry in L of the form (y, z); if not, choose $z \leftarrow U_{k'}$ and add one. Return $\mathcal{SE}.D_z(y)$.

When A returns the challenge plaintext m^*, B computes $c^* = \mathcal{O}(m^*)$ and gives A the challenge value $y^*\|c^*$. B then proceeds to run A, answering queries in the same manner. If B never terminates to play the OWP game, B decides to play the $\mathsf{IND\$\text{-}CCA}$ game and outputs A's decision. Now let P denote the event that A queries $H(x)$ on an x such that $\pi(x) = y^*$. Clearly,

$$\mathbf{Adv}^{\mathsf{ow}}_{B,\Pi}(k) = \Pr[\mathsf{P}] \ .$$

Now, conditioned on $\neg\mathsf{P}$, when B's oracle $\mathcal{O}$ is a random string oracle, $c^* \leftarrow U_\ell$ and B perfectly simulates the random-string world to A. And (still conditioned on $\neg\mathsf{P}$) when B's oracle $\mathcal{O}$ is E_K, B perfectly simulates the ciphertext world to A. Thus, we have that:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cca}}_{B,\mathcal{SE}}(k) &= \Pr[B^{\$,\mathcal{SE}.D_K}(\pi, y) = 1] - \Pr[B^{\mathcal{SE}.E_K,\mathcal{SE}.D_K}(\pi, y) = 1] \\
&= \Pr[A^{\mathcal{E}.D_{SK}}(U_\ell) = 1|\neg\mathsf{P}] - \Pr[A^{\mathcal{E}.D_{SK}}(\mathcal{E}.E(\pi, m^*)) = 1|\neg\mathsf{P}]
\end{aligned}$$

But this gives us

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k) &= \Pr[A^{\mathcal{E}.D_{SK}}(U_\ell) = 1] - \Pr[A^{\mathcal{E}.D_{SK}}(\mathcal{E}.E(\pi, m^*)) = 1] \\
&= \big(\Pr[A^{\mathcal{E}.D_{SK}}(U_\ell) = 1|\neg\mathsf{P}] - \Pr[A^{\mathcal{E}.D_{SK}}(\mathcal{E}.E(\pi, m^*)) = 1|\neg\mathsf{P}]\big)\Pr[\neg\mathsf{P}] \\
&\quad + \big(\Pr[A^{\mathcal{E}.D_{SK}}(U_\ell) = 1|\mathsf{P}] - \Pr[A^{\mathcal{E}.D_{SK}}(\mathcal{E}.E(\pi, m^*)) = 1|\mathsf{P}]\big)\Pr[\mathsf{P}] \\
&\leq \Pr[A^{\mathcal{E}.D_{SK}}(U_\ell) = 1|\neg\mathsf{P}] - \Pr[A^{\mathcal{E}.D_{SK}}(\mathcal{E}.E(\pi, m^*)) = 1|\neg\mathsf{P}] + \Pr[\mathsf{P}] \\
&= \mathbf{Adv}^{\mathsf{cca}}_{B,\mathcal{SE}}(k) + \mathbf{Adv}^{\mathsf{ow}}_{B,\Pi}(k) \\
&\leq \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{SE}}(t', 1, q, l, \mu, k) + \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t', k)
\end{aligned}$$

□

SS-CCA Game

In an adaptive chosen-covertext attack against a public-key stegosystem $\mathcal{S}$, a challenger draws a key pair $(PK, SK) \leftarrow SG(1^k)$, and an adversary W is given PK and allowed oracle access to SD_{SK}. The attacker produces a *challenge hiddentext* m^* and history h^* and is given as a response a sequence of documents $s^* \in D^{\ell(|m^*|)}$. After this, the attacker continues to query SD with the restriction that he may not query $SD(s^*)$. (As always, W is allowed to know the channel distribution $\mathcal{C}$) At the conclusion of the attack, W must guess whether $s^* \leftarrow SE(PK, m^*, h^*)$ or $s^* \leftarrow \mathcal{C}_h^{\ell^*}$. We define the *Steganographic Chosen-Covertext Advantage* of W against $\mathcal{S}$ with respect to $\mathcal{C}$ by

$$\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},\mathcal{W}}(k) = \left|\Pr[W^{SD_{SK}}(PK, SE(PK, m^*, h^*)) = 1] - \Pr[W^{SD_{SK}}(PK, \mathcal{C}_h^{l^*}) = 1]\right| ,$$

where $(m^*, h^*) \leftarrow W^{SD_{SK}}(PK)$ and $(PK, SK) \leftarrow SG(1^k)$. We define the sCCA insecurity of $\mathcal{S}$ with respect to $\mathcal{C}$ by

$$\mathbf{InSec}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C}}(t, q, \mu, l^*, k) = \max_{W \in \mathcal{W}(t,q,\mu,l^*)} \left\{ \mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},\mathcal{W}}(k) \right\} ,$$

where $\mathcal{W}(t, q, \mu, l^*)$ denotes the class of all W running in time t which make at most q oracle queries of μ bits and submit a challenge hiddentext of length at most l^*.

We say that $\mathcal{S}$ is $(t, q, \mu, l, k, \epsilon)$ secure against chosen-covertext attack with respect to $\mathcal{C}$ if

$$\mathbf{InSec}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C}}(t, q, \mu, l, k) \leq \epsilon ,$$

and that $\mathcal{S}$ is secure against chosen-covertext attack with respect to $\mathcal{C}$ (SS-sCCA) if $\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},\mathcal{W}}(k)$ is negligible for all PPT W.

Construction

Our construction of a public-key stegosystem secure against chosen-covertext attack is similar to the construction for the symmetric-key case. We will assume that $\mathcal{E}$ is a public-key IND\$-CCA secure encryption scheme, and $(PK, SK) \leftarrow \mathcal{E}.G(1^k)$. Furthermore, we assume that for any l, $\Pr[SD_{SK}(U_l) \neq \perp] \leq 2^{-t}$ for some $t = poly(k)$. We will also assume that $G : \{0,1\}^k \to \{0,1\}^{k \times lk}$ is a pseudorandom generator.

The the following scheme is secure against chosen-covertext attack:

Construction 5.21. (pCCA Stegosystem)

Procedure `Encode`:	**Procedure** `Decode`:
Input: $m \in \{0,1\}^\ell$, h, PK	**Input:** $s_1, \ldots, s_l$, h, SK
Choose $r \leftarrow U_k$	Let $c = \mathtt{Basic\_Decode}(s_1, \ldots, s_l)$
Let $c = E_{PK}(r \| m)$	Parse $D_{SK}(c)$ as $r \|_k m$.
Let $\vec{r} = G(r)$	Set $\vec{r} = G(r)$.
Output: $\mathtt{DEncode}(c, h, k, \vec{r})$	If $s \neq \mathtt{DEncode}(c, h, k, \vec{r})$ return $\perp$.
	Output: m

Theorem 5.22. *if f is ϵ-biased for $\mathcal{C}$, then*

$$\mathbf{InSec}^{\mathsf{scca}}_{\mathtt{pCCA},\mathcal{C}}(t, q, \mu, l, k) \leq \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t', q, \mu, l, k) + 2^{-t} + \ell(l+k)\epsilon + \mathbf{InSec}^{\mathsf{prg}}_{G}(t', k) ,$$

where $t' \leq t + O(lk)$.

Proof. Choose an arbitrary $W \in \mathcal{W}(t, q, \mu, l)$; let $(PK, SK) \leftarrow G(1^k)$ and let

$$(m^*, h^*) \leftarrow W^{SD_{SK}}(PK) \ .$$

We will bound $\mathbf{Adv}^{\mathsf{scca}}_{W,\mathtt{pCCA},\mathcal{C}}(k)$ by considering the following sequence of hybrid distribution:

- D_1: $\mathcal{C}_{h^*}^{\ell(l+k)}$
- D_2: $\mathtt{DEncode}(U_{\ell(l+k)}, h^*, k, U_{k\times lk})$
- D_3: $\mathtt{DEncode}(U_{\ell(l+k)}, h^*, k, G(U_k))$
- D_4: $\mathtt{DEncode}(E_{PK}(r\|m^*), h^*, k, G(r))$, where $r \leftarrow U_k$

Clearly $\Pr[W^{SD}(D_4) = 1] = \Pr[W^{SD}(SE(PK, m^*, h^*)) = 1]$ and $\Pr[W^{SD}(D_1) = 1] = \Pr[W^{SD}(\mathcal{C}_{h^*}^{\ell(l+k)}) = 1]$. Thus

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{scca}}_{W,\mathtt{pCCA},\mathcal{C}}(k) &= \left|\Pr[W^{SD}(D_4) = 1] - \Pr[W^{SD}(D_1) = 1]\right| \\
&\leq \left|\Pr[W^{SD}(D_2) = 1] - \Pr[W^{SD}(D_1) = 1]\right| \\
&\quad + \left|\Pr[W^{SD}(D_3) = 1] - \Pr[W^{SD}(D_2) = 1]\right| \\
&\quad + \left|\Pr[W^{SD}(D_4) = 1] - \Pr[W^{SD}(D_3) = 1]\right|
\end{aligned}$$

For convenience, we will define the quantity

$$\mathbf{Adv}^{i}_{W}(k) = \left|\Pr[W^{SD}(D_{i+1}) = 1] - \Pr[W^{SD}(D_i) = 1]\right| \ ,$$

and we will proceed to bound $\mathbf{Adv}^{i}_{W}(k)$ for $i \in \{1, 2, 3\}$.

Lemma 5.23. $\mathbf{Adv}^{1}_{W}(k) \leq \ell(l+k)\epsilon$

Proof. This follows from lemma 4.8. □

Lemma 5.24. $\mathbf{Adv}^{2}_{W}(k) \leq \mathbf{InSec}^{\mathsf{prg}}_{G}(t', k)$

Proof. We will construct a PRG adversary A for G such that

$$\mathbf{Adv}^{\mathsf{prg}}_{G,A}(k) = \mathbf{Adv}^{2}_{W}(k) \ .$$

A works as follows: first, A picks a key pair $(PK, SK) \leftarrow G(1^k)$ to use in responding to the queries W makes to SD. A is given as input a string $r \in \{0,1\}^{k \times lk}$ and asked to decide whether $r \leftarrow U_{k \times lk}$ or $r \leftarrow G(U_k)$. Then A can achieve advantage precisely $\mathbf{Adv}^2_W(k)$ by emulating W, responding to its decoding queries using SK, and responding to the challenge hiddentext (m^*, h^*) by drawing $c \leftarrow U_{\ell(l+k)}$ and giving the response $s = \texttt{DEncode}(c, h, k, r)$. If $r \leftarrow U_{k \times lk}$, then $s \leftarrow D_1$, and if $r \leftarrow G(U_k)$, then $s \leftarrow D_2$. Thus A's advantage in distinguishing $G(U_k)$ and $U_{k \times lk}$ is exactly:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prg}}_{A,G}(k) &= |\Pr[A(G(U_k)) = 1] - \Pr[A(U_{k \times lk}) = 1]| \\
&= \left|\Pr[W^{SD}(D_2) = 1] - \Pr[W^{SD}(D_1) = 1]\right| \\
&= \mathbf{Adv}^2_W(k)
\end{aligned}$$

$\square$

Lemma 5.25. $\mathbf{Adv}^3_W(k) \leq \mathbf{InSec}^{\mathsf{cca}}_{\mathcal{E}}(t', \vec{q}, \vec{\mu}, k) + 2^{-t}$

Proof. We will construct an adversary A that plays the chosen-ciphertext attack game against $\mathcal{E}$ with advantage

$$\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k) \geq \mathbf{Adv}^3_W(k) \ .$$

A starts by emulating W to get a challenge hiddentext, responding to decoding queries as follows: on query (s, h), A computes $c = \texttt{Basic\_Decode}(s, h)$; A then uses its decryption oracle to compute $r\|_k m = D_{SK}(c)$. If $c \neq \perp$ and $s = \texttt{DEncode}(c, h, k, G(r))$, A returns m, otherwise A returns $\perp$.

When W generates challenge (m^*, h^*), A chooses $r^* \leftarrow U_k$ and outputs the challenge $r^*\|m^*$. A is given the challenge ciphertext c^* and returns

$$s^* = \texttt{DEncode}(c^*, h^*, k, G(r^*))$$

to W.

A continues to emulate W, responding to queries as follows: on decoding query (s, h), A computes $c = \texttt{Basic\_Decode}(s, h)$; if $c = c^*$ A returns $\perp$, otherwise A uses its decryption oracle to compute $r\|_k m = D_{SK}(c)$. If $c \neq \perp$ and $s = \texttt{DEncode}(c, h, k, G(r))$, A returns m, otherwise A returns $\perp$.

In other words, A simulates running `sCCA.Decode` with its D_{SK} oracle, except that because A is playing the $\mathsf{IND\$\text{-}CCA}$ game, he is not allowed to query D_{SK} on the challenge value c^*: thus a decoding query that has the same underlying ciphertext c^* must be dealt with specially.

Notice that when A is given an encryption of $r^*\|m^*$, he perfectly simulates D_4 to W, that is:

$$\Pr[A^{D_{SK}}(PK, E_{PK}(r^*\|m^*) = 1] = \Pr[W^{SD}(PK, D_4) = 1] \ .$$

This is because when $c^* = E_K(r^*\|m^*)$ then the test $s = \mathtt{DEncode}(c, h, k, G(r))$ would fail anyways.

Likewise, when A is given a random string, he perfectly simulates D_3 to W, *given that c^* is not a valid ciphertext*. Let us denote the event that c^* is a valid ciphertext by V, and the event that a sample from D_3 encodes a valid ciphertext by U; notice that by construction $\Pr[\mathsf{U}] = \Pr[\mathsf{V}]$. We then have that

$$\begin{aligned}
\Pr[A^{D_{SK}}(PK, U_\ell) = 1] &= \Pr[A^{D_{SK}}(PK, U_\ell) = 1|\neg\mathsf{V}] \Pr[\neg\mathsf{V}] \\
&\quad + \Pr[A^{D_{SK}}(PK, U_\ell) = 1|\mathsf{V}] \Pr[\mathsf{V}] \\
&\le \Pr[W^{SD}(PK, D_3) = 1|\neg\mathsf{U}] \Pr[\neg\mathsf{U}] + \Pr[\mathsf{V}] \\
&\le \Pr[W^{SD}(PK, D_3) = 1] + \Pr[\mathsf{V}] \\
&\le \Pr[W^{SD}(PK, D_3) = 1] + 2^{-t} \ ,
\end{aligned}$$

since $\Pr[\mathsf{V}] \le 2^{-t}$ by assumption on $\mathcal{E}$.

Combining the cases, we find that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cca}}_{A,\mathcal{E}}(k) &= \Pr[A^{D_{SK}}(PK, E_{PK}(r^*\|m^*) = 1] - \Pr[A^{D_{SK}}(PK, U_\ell) = 1] \\
&= \Pr[W^{SD}(PK, D_4) = 1] - \Pr[A^{D_{SK}}(PK, U_\ell) = 1] \\
&\ge \Pr[W^{SD}(PK, D_4) = 1] - \Pr[W^{SD}(PK, D_3) = 1] - 2^{-t} \\
&= \mathbf{Adv}^3_W(k) - 2^{-t}
\end{aligned}$$

Which proves the lemma. □

Combining the three lemmata yields the proof of the theorem. □

5.2.2 Authenticated Stegosystems

In the case of public-key steganography, Ward is capable of an even stronger attack than the the CCA attack. For example, the warden can detect the use of steganography by Bob simply by encoding a message, sending it to Bob and watching his reaction: if he reacts consistently with receiving the warden's message, then he is probably decoding messages. Thus the warden's goal should be to detect whether a specific pair, Alice and Bob are communicating steganographically. To protect against such an attack will require that Alice have some secret differentiating herself from the warden: we will allow Alice to publish a "steganographic verification key" which will allow anyone with private key SK to verify that a stegotext generated with the corresponding public key PK was generated by Alice; Alice will keep the "steganographic signature" key secret. In this model, we will define additional attack games to the basic chosen-hiddentext attack: the Chosen Exactly One Attack, and the Chosen Stegotext Attack.

Before we can do so, however, it is necessary to extend the syntax and correctness definitions of a public-key stegosystem to include steganographic signatures.

Definition 5.26. An *authenticated public-key stegosystem* $\mathcal{S}$ is a quadruple of algorithms:

- $\mathcal{S}$.`CodeGen` takes as input a security parameter 1^k and generates a key pair $(PK, SK) \in \mathcal{PK} \times \mathcal{SK}$. When it is clear from the context which stegosystem we are referring to, we will abbreviate $\mathcal{S}$.`Generate` by SG.
- $\mathcal{S}$.`SigGen` (abbreviated SSG when $\mathcal{S}$ is clear from the context) takes as input a security parameter 1^k and generates a key pair $(SVK, SSK) \in \mathcal{SVK} \times \mathcal{SSK}$.
- $\mathcal{S}$.`Encode` (abbreviated SE when $\mathcal{S}$ is clear from the context) takes as input a public key $PK \in \mathcal{PK}$, a stegosigning key $SSK \in \mathcal{SSK}$, a string $m \in \{0,1\}^*$ (the *hiddentext*), and a message history h. As with the symmetric case, we will also assume for our feasibility results that SE has access to a channel oracle for some channel $\mathcal{C}$, which can sample from $\mathcal{C}_h$ for any h.

$SE(PK, SSK, m, h)$ returns a sequence of documents $s_1, s_2, \ldots, s_l$ (the *stegotext*) from the support of $\mathcal{C}_h^l$.

- $\mathcal{S}$.`Decode` (abbreviated SD) takes as input a secret key $SK \in \mathcal{SK}$, a stegoverification key SVK, a sequence of documents $s_1, s_2, \ldots, s_l$, and a message history h.

  $SD(SK, SVK, s, h)$ returns a hiddentext $m \in \{0,1\}^*$ or $\perp$.

Of course, we will also require that an authenticated stegosystem is correct:

Definition 5.27. (Correctness) An authenticated stegosystem $\mathcal{S}$ is *correct* if for every polynomial $p(k)$ there exists a negligible $\nu(k)$ such that

$$\forall m \in \{0,1\}^{p(k)}, h \in D^* : \ \Pr[SD(SK, SVK, SE(PK, SSK, m, h), h) = m] \geq 1 - \mu(k)$$

where the probability is over the choices $(PK, SK) \leftarrow SG(1^k)$, $(SVK, SSK) \leftarrow SSG(1^k)$, and any coin tosses of SE, SD, SG, SSG and the oracle to $\mathcal{C}_h$.

We will model an active warden attacking a stegosystem as an efficient oracle machine which plays the following oracle-distinguishing game:

1. W is given access to an oracle which samples documents from the marginal channel distributions $\mathcal{C}_{A \to B,h}$ for any history h. (This oracle allows W to learn the *covertext* distribution on all communications.)

2. W is given access to a second oracle which is either ST_{atk} or CT_{atk}. The oracle ST (for StegoText) will model the case in which the pair Alice and Bob are communicating steganographically, while the oracle CT (for CoverText) will model the case in which they are not. The exact distributions over $ST_{\text{atk}}, CT_{\text{atk}}$ vary depending on the attack model, atk. Below we will specify these distributions for $\mathsf{atk} \in \{\text{cxo}, \text{csa}\}$. Both oracles respond to the null query with the public and verification keys generated by SG and SSG.

3. In the end, W outputs a bit.

We define the *advantage of W against stegosystem S* over channel $\mathcal{C}$ in attack model atk by

$$\mathbf{Adv}^{\mathsf{atk}}_{S,\mathcal{C},W}(k) = \left|\Pr[W^{ST_{\text{atk}}}(1^k) = 1] - \Pr[W^{CT_{\text{atk}}}(1^k) = 1]\right| \ ,$$

where the warden uses random bits r and the probabilities are also taken over the oracles. Define *the insecurity of S over channel $\mathcal{C}$* by

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

where $\mathcal{W}(t, q, l)$ denotes the set of all adversaries which make at most q oracle queries totaling at most l bits, and running in time at most t.

Definition 5.28. (Steganographic secrecy) A Stegosystem S is called (t, q, l, k, ϵ)-*steganographically secret against* atk ($\mathsf{SS\text{-}ATK}$) for the channel $\mathcal{C}$ if $\mathbf{InSec}^{\mathsf{atk}}_{S,\mathcal{C}}(t, q, l, k) \leq \epsilon$.

For the next construction, we will require the notion of a digital signature scheme that is existentially unforgeable under chosen message attack:

Existentially Unforgeable Digital Signature Schemes.

A digital signature scheme $\mathcal{SG}$ is a triple of probabilistic algorithms:

- $\mathcal{SG}$.`Generate`: takes as input a security parameter 1^k and returns a key pair (VK, SK).
- $\mathcal{SG}$.`Sign`: takes as input a *signing key* SK and a message m and outputs a signature σ.
- $\mathcal{SG}$.`Verify`: takes as input a *verification key* VK, a message m, and a signature σ and outputs a bit.

A signature scheme is sound if $V(VK, m, S(SK, m)) = 1$ for all m and $(SK, VK) \in [G(1^k)]$.

Consider the following game that an adversary A plays against $\mathcal{SG}$: the adversary A is given VK and oracle access to S_{SK}, where $(SK, VK) \leftarrow G(1^k)$. A makes q oracle queries of at most l bits to get back $\{S_{SK}(M_1), ..., S_{SK}(M_q)\}$. A then outputs a pair (M, σ_M). A wins if $M \notin \{M_1, ..., M_q\}$ and $V(VK, M, \sigma_M) = 1$.

Denote the event of A winning the game by $win_{A,\mathcal{SG}}(k)$, Let $\mathcal{A}(t, q, l)$ 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 $\mathsf{EUF\text{-}CMA}$ advantage of A against $\mathcal{SG}$ as

$$\mathbf{Adv}^{\mathsf{cma}}_{A,\mathcal{SG}}(k) = \Pr[win_{A,\mathcal{SG}}(k)] \ .$$

Define the insecurity of $\mathcal{SG}$ as

$$\mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t, q, l, k) = \max_{A \in \mathcal{A}(t,q,l)} \left\{ \mathbf{Adv}^{\mathsf{cma}}_{\mathcal{SG},A}(k) \right\} \ .$$

We say that $\mathcal{SG}$ is (t, q, l, k, ϵ)-*existentially unforgeable under chosen message attack* if $\mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t, q, l, k) \leq \epsilon$. $\mathcal{SG}$ is called *existentially unforgeable under chosen message attack* ($\mathsf{EUF\text{-}CMA}$) if for every PPT A, $\mathbf{Adv}^{\mathsf{cma}}_{\mathcal{SG},A}(k)$ is negligible in k. We note that $\mathsf{EUF\text{-}CMA}$ signature schemes exist if and only if one-way functions exist[47, 55]..

Chosen Exactly-One Attack (CXO).

In a chosen exactly-one attack, Ward may both submit hiddentexts to Alice for encoding, and submit stegotexts to Bob for decoding, but *he is prohibited from accessing both at the same time* (i.e., with the same history). This models a scenario in which Ward can inject messages onto a network, but cannot replace them; for example, this may be the situation if he is connected to the same shared access network as Bob, but does not control the medium. As in all of our attack models, Ward's goal is to determine whether the specific pair of Alice and Bob are communicating steganographically rather than whether Alice or Bob are using steganography at all. In the case that Ward asks Bob to decode stegotexts, his goal is to detect whether Bob is receiving steganography from Alice by watching Bob's reaction to stegotexts (allegedly from Alice) generated by Ward himself. If Bob is not receiving steganography from Alice, he will always output the distinguished message $\perp$, meaning "no reaction." If Bob is receiving steganography from Alice, he will decode the stegotexts.

We formally define the oracle distributions $ST_{\text{cxo}}, CT_{\text{cxo}}$ as follows: First, draw $(PK, SK) \leftarrow SG(1^k)$, and set $\phi = \{\}$. Respond to queries using these programs:

$ST_{\text{cxo}}(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if (b $= \texttt{enc}$) then:
  Sample $s \leftarrow SE(PK, m, h)$
  Set $\phi = \phi \cup \{h\}$
  return s
else
  If $h \in \phi$ return ε
  else return $SD(SK, m, h)$

$CT_{\text{cxo}}(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if (b $= \texttt{enc}$) then:
  Sample $s \leftarrow \mathcal{C}_h^{|SE(PK,m,h)|}$
  Set $\phi = \phi \cup \{h\}$
  return s
else
  If $h \in \phi$ return ε
  else return $\perp$

Note that $\mathbf{InSec}^{\mathsf{cha}}_{S,\mathcal{C}}(t, l, k) \leq \mathbf{InSec}^{\mathsf{cxo}}_{S,\mathcal{C}}(t, 1, l, k)$, since any CHA warden can be emulated by a CXO warden making only $(\texttt{enc}, m, h)$-queries.

SS-CXO is the strongest notion of security that we are able to achieve in the standard model. Since any stegotext encoded by Alice can be thought of as a "challenge stegotext," (Ward's goal is only to detect that it is, in fact, a stegotext rather than a covertext) this condition is somewhat analogous to non-adaptive chosen ciphertext security (IND-CCA1) for public-key encryption.

Let $\mathcal{SG} = (G_S, S, V)$ be a EUF-CMA secure signature scheme, with signature key K_S and verification key K_V, and let $\mathcal{E} = (G, E, D)$ be a IND\$-CPA encryption scheme with public key PK and secret key SK. Let ℓ be the expansion function of $\mathcal{E}$ and let ℓ_σ be the length of signatures generated by $\mathcal{SG}$. Then the following construction yields a SS-CXO secure stegosystem from Alice to Bob, when Alice knows PK, K_S and Bob knows SK, K_V. Assume also that all keys are generated with security parameter k.

Construction 5.29. (Chosen Exactly-One Security)

Procedure `CXO_Encode`:
Input: m, h, PK, K_S
Let $c = E_{PK}(m, S_{K_S}(h, m))$
Output: `Basic_Encode`(c, h, k)

Procedure `CXO_Decode`:
Input: $s_1, \ldots, s_l$, h, SK, K_V
Let $c =$ `Basic_Decode`$(s_1, \ldots, s_l)$
Let $(m, \sigma) = D_{SK}(c)$
If $V(K_V, (h, m), \sigma) = 0$ then set $m = \perp$
Output: m

Theorem 5.30. *Assume f is ϵ-biased on $\mathcal{C}_h$ for all h. Then*

$$\mathbf{InSec}^{\mathsf{cxo}}_{\mathtt{CXO},\mathcal{C}}(t, q, l, k) \leq \mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t + O(kl), q, l, k) + \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kl), q, l + q\ell_\sigma, k) + \ell(l + q\ell_\sigma)\epsilon \ .$$

Proof. Informally, we will consider the hybrid oracle H which answers encoding queries using `CXO_Encode` and answers all decoding queries with $\perp$. Distinguishing this hybrid from ST_{cxo} equates to distinguishing `CXO_Decode` from the constant oracle $\perp$ on some history h for which no query of the form $(\texttt{enc}, *, h)$ has been made. This can only happen if a decoding query contains a signature on a (m, h) pair which was never signed by `CXO_Encode` (because no encoding queries were ever made with the history h). So, intuitively, distinguishing between H and ST_{cxo} requires forging a signature. Similarly, since both H and CT_{cxo} answer all `dec` queries by $\perp$, distinguishing between them amounts to a chosen-hiddentext attack, which by Lemma 4.10 would give an IND\$-CPA attacker for $\mathcal{E}$. The result follows by the triangle inequality.

More formally, Let $W \in \mathcal{W}(t, q, l)$. We will show that W must either forge a signature or distinguish the output of E from random bits. We will abuse notation slightly and denote $W^{ST_{\text{cxo}}}$ by $W^{SE,SD}$, and $W^{CT_{\text{cxo}}}$ by $W^{\mathcal{C},\perp}$. Then we have that

$$\mathbf{Adv}^{\mathsf{cxo}}_{\mathtt{CXO},\mathcal{C},W}(k) = \left|\Pr[W^{SE,SD} = 1] - \Pr[W^{\mathcal{C},\perp} = 1]\right| \ .$$

Consider the "hybrid" distribution which results by answering encoding queries using `CXO_Encode` but answering all decoding queries with $\perp$. (We denote this oracle by $(SE, \perp)$)

We construct a EUF-CMA adversary $\mathbf{A}_f$ which works as follows: given K_V, and a signing oracle for K_S, choose $(PK, SK) \leftarrow G_E(1^k)$; use the signing oracle and E_{PK}, D_{SK} to emulate `CXO_Encode` and `CXO_Decode` to W. If W ever makes a query to `CXO_Decode` which does not return $\perp$ then $\mathbf{A}_f$ halts and returns the corresponding $((m, h), \sigma)$ pair, otherwise $\mathbf{A}_f$ runs until W halts and returns $(0, 0)$. If we let F denote the event that $W^{SE,SD}$ submits a valid decoding query to `CXO_Decode`, then we have that $\mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f) = \Pr[F]$.

We also construct a IND\$-CPA adversary $\mathbf{A}_d$ which works as follows: given an encryption oracle, choose $(K_S, K_V) \leftarrow G_S(1^k)$, use K_S and the encryption oracle to emulate `CXO_Encode` to W, and respond to any decoding queries with $\perp$. $\mathbf{A}_d$ returns the output of W. Note that $\mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l + q\ell_\sigma)\epsilon \geq \left|\Pr[W^{SE,\perp} = 1] - \Pr[W^{\mathcal{C},\perp} = 1]\right|$, which follows from Theorem 4.11.

Then we have the following inequalities:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cxo}}_{\mathsf{CXO},\mathcal{C},W}(k) &= \left|\Pr[W^{SE,SD}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right| \\
&\le \left|\Pr[W^{SE,SD}=1] - \Pr[W^{SE,\perp}=1]\right| \\
&\quad + \left|\Pr[W^{SE,\perp}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right| \\
&\le \left|\Pr[W^{SE,SD}=1] - \Pr[W^{SE,\perp}=1]\right| + \mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l+q\ell_\sigma)\epsilon \\
&\le \mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f) + \mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l+q\ell_\sigma)\epsilon
\end{aligned}$$

Where the last line follows because (let D denote the event $W^{SE,SD}=1$, and notice that $D|\overline{F} \equiv W^{SE,\perp}=1$):

$$\begin{aligned}
\left|\Pr[D] - \Pr[W^{SE,\perp}=1]\right| &= \left|\Pr[D|F]\Pr[F] + \Pr[D|\overline{F}]\Pr[\overline{F}] - (\Pr[W^{SE,\perp}=1])\right| \\
&= \left|\Pr[D|F]\Pr[F] + \Pr[W^{SE,\perp}=1](1-\Pr[F])\right. \\
&\quad \left. - (\Pr[W^{SE,\perp}=1])\right| \\
&= \left|\Pr[F](\Pr[D|F] - \Pr[D|\overline{F}])\right| \\
&\le \Pr[F] \\
&= \mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f)
\end{aligned}$$

The theorem follows by the definition of insecurity, since both $\mathbf{A}_d$ and $\mathbf{A}_f$ run in time at most $t + O(kl)$. □

Chosen-Stegotext Attacks

We now introduce a security condition analogous to adaptive chosen ciphertext security for encryption schemes (IND-CCA2), SS-CSA. We will construct a stegosystem which is SS-CSA-secure in the Random Oracle model, for any channel which is *efficiently sampleable*: that is, for which there exists a PPT C such that for all h, $\mathsf{C}(h; U_k)$ and $\mathcal{C}_h$ are indistinguishable.

We formally define the oracle distributions $ST_{\text{csa}}, CT_{\text{csa}}$ as follows: First, draw $(PK, SK) \leftarrow SG(1^k)$, and set $\phi = \{\}$. Respond to queries using these programs:

ST_{csa} $(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if(b $=$ $\texttt{enc}$) then:
  Sample $s \leftarrow SE(PK, m, h)$
  Set $\phi = \phi \cup \{(s, h)\}$
  return s
else
  If $(m, h) \in \phi$ return ε
  else return $SD(SK, m, h)$

CT_{csa} $(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if(b $=$ $\texttt{enc}$) then:
  Sample $s \leftarrow \mathcal{C}_h^{|SE(PK,m,h)|}$
  Set $\phi = \phi \cup \{(s, h)\}$
  return s
else
  If $(m, h) \in \phi$ return ε
  else return $\perp$

Thus, in a chosen-stegotext attack, Ward may ask Alice to encode any (message, history) pair of his choosing, as often as he likes, and may query Bob on any pair (s, h) where s was not a result of an encoding query for history h. Notice that if Ward can ask Bob to decode any message output by Alice for the same history it was encoded with, he can detect the use of steganography between Alice and Bob; this is why we do not allow Ward to query Bob on such stegotexts. This restriction is roughly analogous to the standard restriction that an adaptive chosen-ciphertext attacker may not query his decryption oracle on the challenge ciphertext. Advantage and insecurity for SS-CSA are defined analogously to SS-CXO, except that we count encoding and decoding queries separately (as q_e and q_d) as well as counting the number of queries made to random oracles.

Construction.

We assume that π_A, π_B are elements of trapdoor one-way permutation family Π_k, where Alice knows π_A^{-1} and Bob knows π_B^{-1}. In addition, we assume all parties have access to random oracles $F : \{0,1\}^* \rightarrow \{0,1\}^k$, $G : \{0,1\}^* \rightarrow \{0,1\}^k$, $H_1 : \{0,1\}^k \rightarrow \{0,1\}^*$, and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$. The following construction slightly modifies techniques from [9], using the random oracles H_1 and H_2 with π_B to construct a pseudorandom non-malleable encryption scheme and the oracle F in conjunction with π_A to construct a strongly unforgeable signature scheme.

Construction 5.31. (Chosen Stegotext Security)

Procedure $\texttt{CSA\_Encode}^{F,G,H}$:	**Procedure** $\texttt{CSA\_Decode}^{F,G,H}$:
Input: $m_1 \cdots m_\ell$, h, π_A^{-1}, π_B	**Input:** $s_1, \ldots, s_l$, h, π_A, π_B^{-1}
Choose $r \leftarrow U_k$	Let $c = \texttt{Basic\_Decode}(s_1, \ldots, s_l)$
Let $\sigma = \pi_A^{-1}(F(r, m, h))$	Parse c as $y\|\|e\|\|\tau$.
Let $e = H_1(r) \oplus (m, \sigma)$	Set $r = \pi_B^{-1}(y)$.
Let $\tau = H_2(r, m, h)$	If $s \neq \texttt{UEncode}^G(c, r, h)$ return $\perp$.
Let $y = \pi_B(r)$	Let $(m, \sigma) = e \oplus H_1(r)$
Let $c = y\|\|e\|\|\tau$	If $\tau \neq H_2(r, m, h)$ return $\perp$.
Output: $\texttt{UEncode}^G(c, r, h)$	If $\pi_A(\sigma) \neq F(r, m, h)$ return $\perp$.
	Output: m

Procedure $\texttt{UEncode}^G$:
Input: $c \in \{0,1\}^l$, $r \in \{0,1\}^k$, h
for $i = 1 \ldots l$ do
    Let $j = 0$
    repeat:
        set $s_i = \texttt{C}(h; G(h, r, c, j))$
        increment j
    until $f(s_i) = c_i$ OR $(j > k)$
    set $h = (h, s_i)$
Output: $s_1, s_2, \ldots, s_l$

Theorem 5.32. *If f is ϵ-biased for $\mathcal{C}$, then*

$$\mathbf{InSec}^{\mathsf{csa}}_{\mathsf{CSA},\mathcal{C}}(t, \vec{q}, l, k) \leq (2q_e + q_F)\mathbf{InSec}^{\mathsf{ow}}_{\pi}(t', k) + (l + 3q_e k)\epsilon + (q_e^2 + 2q_d)/2^k ,$$

where $t' \leq t + (q_G + q_F + q_{H_1} + q_{H_2})(q_e + q_d)T_\pi + k(l + 3q_e k)T_{\mathtt{C}}$, T_π is the time to evaluate members of π, and $T_{\mathtt{C}}$ is the running time of $\mathtt{C}$.

Proof. Intuitively, this stegosystem is secure because the encryption scheme employed is non-malleable, the signature scheme is strongly unforgeable, and each triple of hiddentext, history, and random-bits has a unique valid stegotext, which contains a signature on (m, h, r). Thus any adversary making a valid decoding query which was not the result of an encoding query can be used to forge a signature for Alice — that is, invert the one-way permutation π_A.

We define the following sequence of hybrid oracle distributions:

1. $\mathsf{P0}(b, m, h) = CT_{\text{csa}}$, the covertext oracle.

2. $\mathsf{P1}(b, m, h)$ responds to $\texttt{dec}$ queries as in $\mathsf{P0}$, and responds to $\texttt{enc}$ queries using $\texttt{CSA\_Encode}^{F,G,H}$ but with calls to $\texttt{UEncode}^G$ replaced by calls to $\texttt{Basic\_Encode}$.

3. $\mathsf{P2}(b, m, h)$ responds to `dec` queries as in $\mathsf{P1}$, and responds to `enc` queries using $\texttt{CSA\_Encode}^{F,G,H}$.

4. $\mathsf{P3}(b, m, h) = ST_{\text{csa}}$, the stegotext oracle.

We are given a CSA attacker $W \in \mathcal{W}(t, q_e, q_d, q_F, q_H, q_{H_1}, q_{H_2}, l)$ and wish to bound his advantage. Notice that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{csa}}_{\mathtt{CSA},\mathcal{C},W}(k) \le\ & |\Pr[W^{P0}(1^k) = 1] - \Pr[W^{P1}(1^k)]| + \\ & |\Pr[W^{P1}(1^k) = 1] - \Pr[W^{P2}(1^k) = 1]| + \\ & |\Pr[W^{P2}(1^k) = 1] - \Pr[W^{P3}(1^k) = 1]| \ .\end{aligned}$$

Hence, we can bound the advantage of W by the sum of its advantages in distinguishing the successive hybrids. For hybrids P, Q we will denote this advantage by $\mathbf{Adv}^{\mathsf{P},\mathsf{Q}}_W(k) = |\Pr[W^{\mathsf{P}}(1^k) = 1] - \Pr[W^{\mathsf{Q}}(1^k) = 1]|$.

Lemma 5.33. $\mathbf{Adv}^{\mathsf{P0},\mathsf{P1}}_W(k) \le q_e \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t', k) + 2^{-k}(q_e^2/2 - q_e/2) + (l + 3q_e k)\epsilon$

Proof. Assume WLOG that $\Pr[W^{P1}(1^k) = 1] > \Pr[W^{P0}(1^k) = 1]$. Let E_r denote the event that, when W queries $\mathsf{P1}$, the random value r never repeats, and let E_q denote the event that W never makes random oracle queries of the form $H_1(r)$ or $H_2(r, *, *)$ for an r used by $\texttt{CSA\_Encode}^{F,G,H}$, and let $E \equiv E_r \wedge E_q$. Then:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{P0},\mathsf{P1}}_W(k) &= \Pr[W^{P1}(1^k) = 1] - \Pr[W^{P0}(1^k) = 1] \\ &= \Pr[W^{P1}(1^k) = 1|E](1 - \Pr[\overline{E}]) + \Pr[W^{P1}(1^k) = 1|\overline{E}] \Pr[\overline{E}] \\ &\quad - \Pr[W^{P0}(1^k) = 1] \\ &= \Pr[\overline{E}] \left(\Pr[W^{P1}(1^k) = 1|\overline{E}] - \Pr[W^{P1}(1^k) = 1|E]\right) \\ &\quad + \left(\Pr[W^{P1}(1^k) = 1|E] - \Pr[W^{P0}(1^k) = 1]\right) \\ &\le \Pr[\overline{E}] \quad + (l + 3q_e k)\epsilon \\ &\le \Pr[\overline{E_r}] + \Pr[\overline{E_q}] + (l + 3q_e k)\epsilon \\ &\le 2^{-k}\frac{q_e(q_e - 1)}{2} + \Pr[\overline{E_q}] + (l + 3q_e k)\epsilon \ ,\end{aligned}$$

because if r never repeats and W never queries $H_1(r)$ or $H_2(r,*,*)$ for some r used by $\texttt{CSA\_Encode}^{F,G,H}$, then W cannot distinguish between the ciphertexts passed to $\texttt{Basic\_Encode}$ and random bit strings.

It remains to bound $\Pr[\overline{E_q}]$. Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_B which is given a value $\pi_B(x)$ and uses W in an attempt to find x, so that $\mathbf{A}$ succeeds with probability at least $(1/q_e)\Pr[\overline{E_q}]$. $\mathbf{A}$ picks (π_A, π_A^{-1}) from Π_k and i uniformly from $\{1, \dots, q_e\}$, and then runs W answering all its oracle queries as follows:

- enc queries are answered as follows: on query $j \neq i$, respond using the program for $\texttt{CSA\_Encode}^{F,G,H}$ with calls to $\texttt{UEncode}^G$ replaced by calls to $\texttt{Basic\_Encode}$. On the i-th query respond with $s = \texttt{Basic\_Encode}(\pi_B(x)||e_1||\tau_1, h)$ where $e_1 = h_1 \oplus (m, \sigma_1)$ and h_1, σ_1, τ_1 are chosen uniformly at random from the set of all strings of the appropriate length ($|e_1| = |m| + k$ and $|\tau_1| = k$), and set $\phi = \phi \cup \{(s, h)\}$.
- dec queries are answered using CT_{csa}.
- Queries to G, F, H_1 and H_2 are answered in the standard manner: if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length. If a query contains a value r for which $\pi_B(r) = \pi_B(x)$, halt the simulation and output r.

It should be clear that $\Pr[\mathbf{A}(\pi_B(x)) = x] \geq \frac{1}{q_e}(\Pr[\overline{E_q}])$. □

Lemma 5.34. $\mathbf{Adv}_W^{\mathsf{P1},\mathsf{P2}}(k) \leq q_e \mathbf{InSec}_\Pi^{\mathsf{ow}}(t', k) + 2^{-k}(q_e^2/2 - q_e/2)$

Proof. Assume WLOG that $\Pr[W^{P2}(1^k) = 1] > \Pr[W^{P1}(1^k) = 1]$. Denote by E_r the event that, when answering queries for W, the random value r of $\texttt{CSA\_Encode}^{F,G,H}$ never repeats, and by E_q the event that W never queries $G(*, r, \pi_B(r)||*, *)$ for some

r used by $\texttt{CSA\_Encode}^{F,G,H}$, and let $E \equiv E_r \wedge E_q$. Then:

$$\begin{aligned}
\mathbf{Adv}_W^{\mathsf{P1,P2}}(k) =& \Pr[W^{P2}(1^k) = 1] - \Pr[W^{P1}(1^k) = 1] \\
=& \ \big(\Pr[W^{P2}(1^k) = 1|E]\Pr[E] + \Pr[W^{P2}(1^k) = 1|\overline{E}]\Pr[\overline{E}]\big) \\
& - \big(\Pr[W^{P1}(1^k) = 1|E]\Pr[E] + \Pr[W^{P1}(1^k) = 1|\overline{E}]\Pr[\overline{E}]\big) \\
=& \ \Pr[\overline{E}]\big(\Pr[W^{P2}(1^k) = 1|\overline{E}] - \Pr[W^{P1}(1^k) = 1|\overline{E}]\big) \\
\leq& \ \Pr[\overline{E}] \\
\leq& 2^{-k}\frac{q_e(q_e-1)}{2} + \Pr[\overline{E_q}]
\end{aligned}$$

Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_B which is given a value $\pi_B(x)$ and uses W in an attempt to find x. $\mathbf{A}$ picks (π_A, π_A^{-1}) from Π_k and i uniformly from $\{1, \ldots, q_E\}$, and then runs W answering all its oracle queries as follows:

- $\texttt{enc}$ queries are answered as follows: on query $j \neq i$, respond according to $\texttt{CSA\_Encode}^{F,G,H}$. On the i^{th} query respond by computing

  $$s = \texttt{UEncode}^G(\pi_B(x)||e_1||\tau_1, r_1, h) \ ,$$

  where $e_1 = h_1 \oplus (m, \sigma_1)$ and $h_1, \sigma_1, \tau_1, r_1$ are chosen uniformly at random from the set of all strings of the appropriate length ($|e_1| = |m| + k$ and $|\tau_1| = k$), and set $\phi = \phi \cup \{(s, h)\}$.

- $\texttt{dec}$ queries are answered using CT_{csa}.

- Queries to G, F, H_1 and H_2 are answered in the standard manner: if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length. If a query contains a value r for which $\pi_B(r) = \pi_B(x)$, halt the simulation and output r.

It should be clear that $\Pr[\mathbf{A}(\pi_B(x)) = x] \geq \frac{1}{q_e}(\Pr[\overline{E_q}])$. □

Lemma 5.35. $\mathbf{Adv}_W^{\mathsf{P2,P3}}(k) \leq q_F \mathbf{InSec}_\Pi^{\mathsf{ow}}(t', k) + q_d/2^{k-1} + q_e/2^k$

Proof. Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_A which is given a value $\pi_A(x)$ and uses W in an attempt to find x. $\mathbf{A}$ chooses (π_B, π_B^{-1}) from Π_k and i uniformly from $\{1, \ldots, q_F\}$, and then runs W answering all its oracle queries as follows:

- `enc` queries are answered using $\texttt{CSA\_Encode}^{F,G,H}$ except that σ is chosen at random and $F(r, m, h)$ is set to be $\pi_A(\sigma)$. If $F(r, m, h)$ was already set, fail the simulation.
- `dec` queries are answered using $\texttt{CSA\_Decode}^{F,G,H}$, with the additional constraint that we reject any stegotext for which there hasn't been an oracle query of the form $H_2(r, m, h)$ or $F(r, m, h)$.
- Queries to G, F, H_1 and H_2 are answered in the standard manner (if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length) except that the i-th query to F is answered using $\pi_A(x)$.

$\mathbf{A}$ then searches all the queries that W made to the decryption oracle for a value σ such that $\pi_A(\sigma) = \pi_A(x)$. This completes the description of $\mathbf{A}$.

Notice that the simulation has a small chance of failure: at most $q_e/2^k$. For the rest of the proof, we assume that the simulation doesn't fail. Let E be the event that W makes a decryption query that is rejected in the simulation, but would not have been rejected by the standard $\texttt{CSA\_Decode}^{F,G,H}$. It is easy to see that $\Pr[E] \leq q_d/2^{k-1}$. Since the only way to differentiate $\mathsf{P3}$ from $\mathsf{P2}$ is by making a decryption query that $\mathsf{P3}$ accepts but $\mathsf{P2}$ rejects, and, conditioned on $\overline{E}$, this can only happen by inverting π_A on a some $F(r, m, h)$, we have that:

$$\mathbf{Adv}_W^{\mathsf{P2},\mathsf{P3}}(k) \leq q_F \mathbf{InSec}_\Pi^{\mathsf{ow}}(t', k) + q_d/2^{k-1} + q_e/2^k$$

$\square$

The theorem follows, because:

$$\begin{aligned}
\mathbf{InSec}^{\mathsf{csa}}_{\mathsf{CSA},\mathcal{C}}(t,\vec{q},l,k) &\leq \mathbf{Adv}^{\mathsf{csa}}_{\mathsf{CSA},\mathcal{C},W_{max}}(k) \\
&\leq \mathbf{Adv}^{\mathsf{P0},\mathsf{P1}}_{W}(k) + \mathbf{Adv}^{\mathsf{P1},\mathsf{P2}}_{W}(k) + \mathbf{Adv}^{\mathsf{P2},\mathsf{P3}}_{W}(k) \\
&\leq q_e \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + \frac{q_e^2 - q_e}{2^{k+1}} + (l + 3q_e k)\epsilon + \mathbf{Adv}^{\mathsf{P1},\mathsf{P3}}_{W}(k) \\
&\leq 2q_e \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + 2^{-k}(q_e^2 - q_e) + (l + 3q_e k)\epsilon + \mathbf{Adv}^{\mathsf{P2},\mathsf{P3}}_{W}(k) \\
&\leq (2q_e + q_F)\mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + 2^{-k}(q_e^2 + 2q_d) + (l + 3q_e k)\epsilon
\end{aligned}$$

□

We conjecture that the *cryptographic* assumptions used here can be weakened; in particular, a random oracle is not necessary given a public-key encryption scheme which satisfies IND\$-CPA and is non-malleable, and a signature scheme which is *strongly unforgeable*. However, we are unaware of an encryption scheme in the standard model satisfying this requirement: nonmalleable encryption schemes following the Naor-Yung paradigm [23, 42, 48, 56] are easily distinguishable from random bits, and the schemes of Cramer and Shoup [20, 21] all seem to generate ciphertexts which are elements of recognizable subgroups. Furthermore, it seems challenging to prevent our motivating attack without assuming the ability to efficiently sample the channel.

5.3 Relationship between robustness and integrity

In this section, we define the notion of a nontrivial relation R and show that if a stegosystem is substitution robust with respect to any nontrivial R then it is insecure against both chosen-covertext and chosen-stegotext attacks. This result implies that *no stegosystem can be simultaneously (nontrivially) secure against disrupting and distinguishing active adversaries.*

We first must define what makes an admissible bounding relation R nontrivial. Suppose R is efficiently computable but has the property that for every efficient A,

$$\Pr_{d \leftarrow \mathcal{C}_{h_A}}[d' = A(1^k, d) \wedge d' \neq d \wedge (d, d') \in R]$$

is negligible. Then *any* steganographically secret stegosystem is trivially robust against R, because no efficient adversary can produce a related stegotext $s_W \neq \sigma$ (except with negligible probability) in the substitution attack game; and thus the decoding of s_W will be s, except with negligible probability. Thus in order for robustness of a stegosystem to be "interesting" we will require that this is not the case.

Definition 5.36. If R is admissible for $\mathcal{C}$ then R is ρ-nontrivial for $\mathcal{C}$ if there is a PPT $\mathbf{A}$ and a history $h_{\mathbf{A}}$ such that

$$\Pr_{d \leftarrow \mathcal{C}_{h_{\mathbf{A}}}}[d' = \mathbf{A}(1^k, d) \wedge d' \neq d \wedge (d, d') \in R] \geq \rho(k) \ .$$

We say that R is non-trivial for $\mathcal{C}$ if it is $\rho(k)$-nontrivial for some $\rho(k) > 1/poly(k)$.

Suppose the stegosystem $\mathcal{S}$ is substitution robust against the nontrivial relation R. Consider the following attacker $W_{\mathbf{A}}$. $W_{\mathbf{A}}$ first selects a *challenge hiddentext* $m_W \leftarrow U_l$ and requests the encoding of m_W under history $h_{\mathbf{A}}$. (In the CSA game, W queries its oracle with $(\mathtt{enc}, m_W, h_{\mathbf{A}})$; in the sCCA game, $W_{\mathbf{A}}$ returns $(m_W, h_{\mathbf{A}})$ as the challenge ciphertext). $W_{\mathbf{A}}$ receives the sequence $\sigma_1, \ldots, \sigma_\ell$ as a response. $W_{\mathbf{A}}$ then computes $s_1 = \mathbf{A}(1^k, \sigma_1)$, attempting to find a $s_1 \neq \sigma_1$ such that $(\sigma_1, s_1) \in R$. If $\mathbf{A}$ is successful, $W_{\mathbf{A}}$ queries its decoding oracle on the sequence $s = s_1, \sigma_2, \ldots, \sigma_\ell$. If the response to this query is m_W, $W_{\mathbf{A}}$ returns 1, otherwise $W_{\mathbf{A}}$ returns 0.

Intuitively, whether this attack is against a CSA or sCCA oracle, it has a significant advantage because when the sequence $\sigma_1, \ldots, \sigma_\ell$ is a stegotext, then the response to the decoding query will be m (because $\mathcal{S}$ is robust); but when it is a covertext, the probability of decoding to m should be low (again because $\mathcal{S}$ is robust). We will now formalize this intuition.

Theorem 5.37.

$$\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},W_{\mathbf{A}}}(k) \geq \rho(k) - \mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_A, 1, l, k) - \mathbf{Fail}^{R}_{\mathcal{S}}(t_{\mathbf{A}}, 0, 0, l, k) - 2^{-l}$$

Proof. Recall that

$$\mathbf{Adv}^{\mathsf{scca}}_{\mathcal{S},\mathcal{C},W_{\mathbf{A}}}(k) = \Pr[W^{SD}(SE(m_W)) = 1] - \Pr[W^{SD}(\mathcal{C}_h^\ell) = 1] \ .$$

Let us first bound $\Pr[W^{SD}(\mathcal{C}_h^\ell) = 1]$. Recall that $W^{SD}(\sigma) = 1$ when

$$SD(s_1, \sigma_2, \ldots, \sigma_\ell) = m_W \ .$$

Let $m_s = SD(s)$; then since s is chosen independently of m_W, and m_W is chosen uniformly from $\{0,1\}^l$, we have that $\Pr[m_s = m_W] \leq 2^{-l}$. Thus

$$\Pr[W^{SD}(\mathcal{C}_h^\ell) = 1] \leq 2^{-l} \ .$$

Let SR denote the event that in the sCCA game played against stegotext, $s_1 \neq \sigma_1 \wedge (\sigma_1, s_1) \in R$. Now notice that

$$\Pr[W^{SD}(SE(m_W)) = 1] \geq \Pr[W^{SD}(SE(m_W)) = 1 | \mathsf{SR}] \Pr[\mathsf{SR}] \ .$$

Because W returns 1 when $SD(s) = m_W$ and s obeys R, we must have that

$$\Pr[W^{SD}(SE(m_W)) \neq 1 | \mathsf{SR}] \leq \mathbf{Fail}_{\mathcal{S}}^R(t_{\mathbf{A}}, 0, 0, l, k) \ ,$$

by the definition of $\mathbf{Fail}_{\mathcal{S}}^R(t_{\mathbf{A}}, 0, 0, l, k)$.

Also, notice that we can exhibit an efficient $\mathsf{SS\text{-}CHA}$ adversary W_ρ against $\mathcal{S}$ such that

$$\mathbf{Adv}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C},W_\rho}(k) \geq \rho(k) - \Pr[\mathsf{SR}] \ .$$

W_ρ works by requesting the encoding of a uniformly chosen message $m^* \leftarrow U_k$ under history $h_{\mathbf{A}}$ to get a sequence starting with $\sigma^* \in D$; W_ρ then computes $s^* \leftarrow \mathbf{A}(1^k, \sigma^*)$ and returns 1 if $(s^* \neq \sigma^*) \wedge (\sigma^*, s^*) \in R$. When $\sigma^* \leftarrow \mathcal{C}_{h_{\mathbf{A}}}$ we have by assumption that

$$\Pr[W_\rho(\mathcal{C}_{h_{\mathbf{A}}}) = 1] \geq \rho(k) \ ,$$

whereas

$$\Pr[W_\rho(SE(m^*)) = 1] = \Pr[\mathsf{SR}] \ ,$$

by construction. Since W_ρ runs in the time it takes to run $\mathbf{A}$ and makes 1 encoding query of k bits, we have that

$$\begin{aligned}
\mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_{\mathbf{A}}, 1, l, k) \geq{}& \mathbf{Adv}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C},W_\rho}(k) \\
&= \Pr[W_\rho(\mathcal{C}_{h_{\mathbf{A}}}) = 1] - \Pr[W_\rho(SE(m^*)) = 1] \\
&\geq \rho(k) - \Pr[\mathsf{SR}]
\end{aligned}$$

Which by rearranging of terms gives us:

$$\Pr[\mathsf{SR}] \geq \rho(k) - \mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_A, 1, l, k) \ .$$

Combining these results, we get that

$$\begin{aligned}\Pr[W^{SD}(SE(m_W)) = 1] &\geq \Pr[W^{SD}(SE(m_W)) = 1|\mathsf{SR}] \Pr[\mathsf{SR}] \\ &\geq (1 - \mathbf{Fail}^{R}_{\mathcal{S}}(t_{\mathbf{A}}, 0, 0, l, k)) \Pr[\mathsf{SR}] \\ &\geq (1 - \mathbf{Fail}^{R}_{\mathcal{S}}(t_{\mathbf{A}}, 0, 0, l, k))(\rho(k) - \mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_{\mathbf{A}}, 1, l, k)) \\ &\geq \rho(k) - \mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_{\mathbf{A}}, 1, l, k) - \mathbf{Fail}^{R}_{\mathcal{S}}(t_{\mathbf{A}}, 0, 0, l, k))\end{aligned}$$

And thus by definition of advantage and insecurity, the theorem follows. □

Theorem 5.38.

$$\mathbf{Adv}^{\mathsf{csa}}_{\mathcal{S},\mathcal{C},W_{\mathbf{A}}}(k) \geq (1 - \mathbf{Fail}^{R}_{\mathcal{S}}(t_{\mathbf{A}}, 0, 0, l, k))(\rho(k) - \mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S},\mathcal{C}}(t_{\mathbf{A}}, 1, l, k))$$

Proof. Recall that

$$\mathbf{Adv}^{\mathsf{csa}}_{\mathcal{S},\mathcal{C},W_{\mathbf{A}}}(k) = \Pr[W_{\mathbf{A}}^{ST_{\mathrm{csa}}}(1^k) = 1] - \Pr[W_{\mathbf{A}}^{CT_{\mathrm{csa}}}(1^k) = 1] \ .$$

It is easy to see that $\Pr[W_{\mathbf{A}}^{CT_{\mathrm{csa}}}(1^k) = 1] = 0$, since querying $CT_{\mathrm{csa}}(\mathtt{enc}, s, h_{\mathbf{A}})$ will always result in $\perp$ or ε, and never m_W. The lower bound for $\Pr[W_{\mathbf{A}}^{ST_{\mathrm{csa}}}(1^k) = 1]$ is proven identically to the stegotext case in the previous proof. □

Chapter 6

Maximizing the Rate

Intuitively, the *rate* of a stegosystem is the number of bits of hiddentext that a stegosystem encodes per document of covertext. Clearly, for practical use a stegosystem should have a relatively high rate, since it may be impractical to send many documents to encode just a few bits. Thus an important question for steganography, first posed by Anderson and Petitcolas [6] is "how much information can be safely encoded by a stegosystem in the channel $\mathcal{C}$?"

A trivial upper bound on the rate of a stegosystem is $\log |D|$. Prior to our work, there were no provably secure stegosystems, and so there was no known lower bound. The rate of the stegosystems defined in the previous chapters is $o(1)$, that is, as the security parameter k goes to infinity, the rate goes to 0. In this chapter, we will address the question of what the optimal rate is for a (universal) stegosystem. We first formalize the definition of the rate of a universal stegosystem. We will then tighten the trivial upper bound by giving a rate MAX such that any universal stegosystem with rate exceeding MAX is insecure. We will then give a matching lower bound by exhibiting a provably secure stegosystem with rate $(1 - o(1))MAX$. Finally we will address the question of what rate a robust stegosystem may achieve.

6.1 Definitions

We concern ourselves with the rate of a universal blockwise, bounded-sample, stegosystem with single-block lookahead.

A *universal stegosystem* $\mathcal{S}$ accepts an oracle for the channel $\mathcal{C}$ and is secure against chosen-hiddentext attack with respect to $\mathcal{C}$ as long as $\mathcal{C}$ does not violate the hardness assumptions $\mathcal{S}$ is based on. Universality is important because typically there is no good description of the marginal distributions on a channel.

A stegosystem is an (h, l, λ)-blockwise stegosystem if it is composed of four functions:

- A *preprocessing function* PE that transforms a hiddentext $m \in \{0,1\}^*$ into a sequence of identically-sized blocks of λ bits.
- A *block encoding function* BE that encodes a block of input bits into a block of l documents.
- A *block decoding function* BD that inverts BE, that is, that transforms a stegotext block into a block of bits.
- A *postprocessing function* PD that inverts PE: that is, transforms a sequence of λ-bit blocks into a hiddentext $m \in \{0,1\}^*$

A blockwise stegosystem computes $SE(K, m, h)$ by first computing $c = PE(K, m)$, then computing $h_0 = h$, and $s_i = BE(K, c_i, h_{i-1})$, $h_i = (s_i, h_{i-1})$. $SD(K, s_{1\dots n}, h)$ is computed by setting $c_i = BD(K, s_i, h_{i-1})$ for $i \in \{1, \dots, n\}$ and computing $m = PD(K, c)$. Because stegotexts are compared to covertexts of the same length as a stegotext, any secure stegosystem can be written as a blockwise stegosystem with a single block.

A (h, l, t)-sample bounded stegosystem uses l-document blocks, draws at most t samples from $\mathcal{C}_h^l$ when encoding a block, and has no other knowledge of $\mathcal{C}_h^l$. Since we require a stegosystem to have bounded running time and to be universal, the running time of $SE(K, m, h)$ is always an upper bound on t. Conversely, if a stegosystem is t-sample bounded, t is a lower bound on the running time of SE.

A (h, l, λ)-blockwise stegosystem has single-block lookahead if $BE(K, c, h)$ draws samples only from $\mathcal{C}_h^l$ and $\mathcal{C}_{h,d}^l$, where $d \in D^l$. Any stegosystem with multi-block lookahead can be transformed into one with single-block lookahead with a larger blocksize.

The *rate* of a stegosystem $\mathcal{S}$ on channel $\mathcal{C}$, $\mathcal{R}_\mathcal{C}(\mathcal{S})$, is given by

$$\mathcal{R}_\mathcal{C}(\mathcal{S}) = \lim_{m \to \infty} E_h \left[\frac{m}{|SE(K, 1^m, \varepsilon)|} \right] ,$$

that is, the number of bits encoded per document by $\mathcal{S}$ on channel $\mathcal{C}$.

6.2 Upper bound

We consider the class $\mathcal{S}(h, l, t)$ of stegosystems which draw at most t samples from $\mathcal{C}_h^l$; we will show two upper bounds on the rate $\mathcal{R}_\mathcal{C}(S)$ for any $S \in \mathcal{S}(h, l, t)$. The first, $MAX_t(S)$ is in terms of the number of samples, t. The second, $MAX_\mathcal{C}(S)$ is in terms of the min entropy $H_\infty(\mathcal{C}_h^l)$ of the channel $\mathcal{C}$. We call the combined upper bound $MAX_\mathcal{C}(h, l, t)$ and define it by

$$MAX_\mathcal{C}(h, l, t) = \min \left\{ MAX_t(S), MAX_\mathcal{C}(S) \right\} .$$

6.2.1 $MAX_t(S)$

For any stegosystem $S \in \mathcal{S}(h, l, t)$, we will show that there exists a channel $\mathcal{C}$ such that S is insecure relative to $\mathcal{C}$ if $\mathcal{R}_\mathcal{C}(S) - \log t$ is any positive constant. Thus it follows that $MAX_t(S) \leq \log t$.

Theorem 6.1.

$$\mathbf{InSec}_{S,\mathcal{C}}^{\mathsf{ss}}(O(t + k), 1, k) \geq 1 - 2^{-c(t,k)} - 2^{-k} - \rho(k) ,$$

where $\rho(k) = \Pr_{m \leftarrow U,K,h}[SD(K, SE(K, m, h), h) \neq m]$ *and* $\mathcal{R}_\mathcal{C}(S) \geq \log t + c(t, k)$.

Proof. The channel $\mathcal{C}$ is defined as follows: to every history h of length k, we associate a polynomial of degree $t + 1$ over $GF(2^k)$ with uniformly chosen coefficients, $p_h(x)$.

To draw from $\mathcal{C}_h$, we draw $x \leftarrow U_k$, compute $y = p_h(x)$ and output $x \| y$. Notice that if $S.SE(K, m, h)$ draws at most t samples $(x_1, y_1), \ldots, (x_t, y_t) \leftarrow \mathcal{C}_h$, and outputs a pair (x', y') such that $x' \notin \{x_1, \ldots, x_t\}$, then $\Pr[y' = p_h(x')] \leq 2^{-k}$. On the other hand, an adversary can draw $t+1$ samples from $\mathcal{C}_h$, and compute p_h by interpolation. Thus when $SE(K, m, h)$ outputs a pair $(x', y') \notin \{(x_1, y_1), \ldots, (x_t, y_t)\}$, an adversary can distinguish between $SE(m, h)$ and $\mathcal{C}_h$ by checking whether $y' = p_h(x')$.

Formally, we define the adversary $A^{\mathcal{O}}$ as follows: on input 1^k, sample a history h of length k by drawing $h_1 \leftarrow \mathcal{C}_\varepsilon$, and $h_i \leftarrow \mathcal{C}_{h_1, \ldots, h_{i-1}}$ for $1 < i \leq k$. Draw $t+1$ samples $(x_1, y_1), \ldots, (x_{t+1}, y_{t+1}) \leftarrow \mathcal{C}_h$. Compute the polynomial p_h by interpolation. Choose $m^* \leftarrow U_{\mathcal{R}_{\mathcal{C}}(S)}$, and draw $(x^*, y^*) \leftarrow \mathcal{O}(m^*, h)$. If $y^* \neq p_h(x^*)$ output 1, otherwise output 0.

Clearly, A runs in time $O(t+k)$. We will now compute the advantage of A. First, notice that given a covertext oracle, A will always output 0:

$$\Pr[A^{CT}(1^k) = 1] = 0 \ .$$

Now, let NS denote the event that $SE(K, m, h)$ draws samples $(x'_1, y'_1), \ldots, (x'_t, y'_t) \leftarrow \mathcal{C}_h$ and outputs a stegotext $(x^*, y^*) \notin \{(x'_1, y'_1), \ldots, (x'_t, y'_t)\}$. Since in this case, $\Pr[y^* = p_h(x^*)] \leq 2^{-k}$, we have that

$$\Pr[A^{ST}(1^k) = 1] \geq \Pr[\mathsf{NS}] - 2^{-k} \ .$$

Thus we only need to give a lower bound on $\Pr[\mathsf{NS}]$ to complete the proof.

Fix a tuple (K, m, h) and consider the set $SD^{-1}_{K,h}(m) = \{s \in D : SD(K, s, h) = m\}$. Since $\mathcal{R}_{\mathcal{C}}(S, h, k) \geq \log t + c(t, k)$, SD partitions D into $t \times 2^{c(t,k)}$ such sets. Then for any fixed set of samples (x'_i, y'_i), the probability over m that $SE(K, m, h)$ has a sample $(x'_i, y'_i) \in SD^{-1}_{K,h}(m)$ is at most $\frac{t}{2^{c(t,k)} t} = 2^{-c(t,k)}$. Let E denote the event that $SE(K, m, h)$ outputs an s^* such that $SD(K, s^*, h) \neq m$. Then

$$\begin{aligned} \Pr[\mathsf{NS}] &\geq \Pr[\forall j, (x'_j, y'_j) \notin SE^{-1}_{K,h}(m)] - \Pr[\mathsf{E}] \\ &\geq 1 - 2^{-c(t,k)} - \rho(k) \ , \end{aligned}$$

which yields the stated bound. $\square$

6.2.2 $MAX_{\mathcal{C}}(S)$

We exhibit a chosen-history, chosen-hiddentext distinguisher for any black box stegosystem (SE, SD) which encodes $\mathcal{R}_{\mathcal{C}}(S) > H_\infty(\mathcal{C}_h^l)$ bits of hiddentext in l documents of stegotext non-negligibly often.

Suppose we have a specific history h such that SE encodes $\ell + 1$ bits by samples from $\mathcal{C}_h^l$, and $H_\infty(\mathcal{C}_h^l) = \ell$. (If such histories occur non-negligibly often then we can find one by sampling from an oracle for SE; if they do not then the rate of the stegosystem doesn't exceed $H_\infty(\mathcal{C}_h^l)$) Since $H_\infty(\mathcal{C}_h^l) = \ell$, we know there is at least one l-document string, s^*, which has probability $2^{-\ell}$ under $\mathcal{C}_h^l$, and no other string has more probability mass than s^*. Now, if SE were deterministic, then we would have that

$$\Pr[SE(m) = s^* : m \leftarrow U_{\ell+1}] \leq 2^{-(\ell+1)} ,$$

by the unique decoding property. Even if SE is randomized, then for any fixed random bits r, we have

$$\Pr[SE(m; r) = s^* : m \leftarrow U_{\ell+1}] \leq 2^{-(\ell+1)} .$$

But then by an averaging argument, there must be some $m^* \in \{0,1\}^{\ell+1}$ such that $\Pr[SE(m^*) = s^*] < 2^{-(\ell+1)}$. In contrast, a covertext oracle CT will have $\Pr[CT(m) = s^*] = 2^{-\ell}$, for any $m \in \{0,1\}^*$. This gap is wide enough to detect with high confidence, given $poly(2^\ell)$ chosen hiddentext samples. And since we are restricted to $\ell = O(\log t)$ by $MAX_t(S)$ this yields a polynomial-time distinguisher between a covertext oracle and a stegotext oracle.

Theorem 6.2. *Let $n = 2^\ell$. Then,*

$$\mathbf{InSec}^{\mathrm{ss}}_{S,\mathcal{C}}(O(n^3), 2n^3, k) \geq 1 - e^{-n/32} - e^{-n/96} - e^{-n/108} - 2ne^{-n/128} ,$$

when $H_\infty(\mathcal{C}_h^l) = \ell$ and $\mathcal{R}_{\mathcal{C}}(S) \geq \ell + 1$.

Proof. We define an adversary W with the stated advantage. $W^{\mathcal{O}}$ executes the following steps:

- W takes n^2 samples from $\mathcal{C}_h^l$. Let $\hat{s}$ be the most commonly occurring l-document in the sample set and let $\hat{p}$ be the number of times $\hat{s}$ occurs in the sample.

- For each document $m \in \{0,1\}^{\ell+1}$, W draws n^2 samples from $\mathcal{O}(m)$. Let $\hat{p}_m$ be the number of occurrences of $\hat{s}$ in the samples from $\mathcal{O}(m)$.

- If for any m, $\hat{p}_m \leq \frac{3}{4}\hat{p}$, W outputs 1. Otherwise W outputs 0.

We will bound the probability that W outputs 1 given a stegotext oracle, and a covertext oracle, respectively.

From the preceding paragraph, we know that when W has a stegotext oracle, there exists an m^* such that $E[\hat{p}_{m^*}] \leq 2^{-\ell+1}$, and we know that $E[\hat{p}] = 2^{-\ell}$. So W will only output 0 if $\hat{p}$ is much smaller then expected, or if $\hat{p}_{m^*}$ is much larger than expected. Specifically, we have:

$$
\begin{aligned}
\Pr[W^{ST}(1^k) = 0] &= \Pr[\hat{p} < \frac{3}{4}2^{-\ell} \wedge \hat{p_{m^*}} \geq \frac{3}{4}\hat{p}] + \Pr[\hat{p} \geq \frac{3}{4}2^{-\ell} \wedge \hat{p_{m^*}} \geq \frac{3}{4}\hat{p}] \\
&\leq \Pr[\hat{p} < \frac{3}{4}2^{-\ell}] + \Pr[\hat{p_{m^*}} \geq \frac{3}{4}\hat{p} | \hat{p} \geq \frac{3}{4}2^{-\ell}] \Pr[\hat{p} \geq \frac{3}{4}2^{-\ell}] \\
&\leq \Pr[\hat{p} < \frac{3}{4}2^{-\ell}] + \Pr[\hat{p_{m^*}} \geq \frac{3}{4}\hat{p} | \hat{p} \geq \frac{3}{4}2^{-\ell}] \\
&= \Pr[\hat{p} < \frac{3}{4}2^{-\ell}] + \Pr[\hat{p_{m^*}} \geq \frac{9}{8}2^{-\ell+1}] \\
&\leq e^{-n/32} + e^{-n/96}
\end{aligned}
$$

Where the last line follows by multiplicative Chernoff bounds. Thus we have

$$
\Pr[W^{ST}(1^k) = 1] \geq 1 - e^{-n/32} + e^{-n/96} \ .
$$

We know that when W has a covertext oracle, it should be the case that for every $m \in \{0,1\}^*$, $E[\hat{p_m}] = 2^{-\ell}$. Thus W should only output 1 when $\hat{p}$ is much larger than expected, or some $\hat{p_m}$ is much smaller than its expectation. Specifically, we have that

$$
\begin{aligned}
\Pr[W^{CT}(1^k) = 1] &= \Pr[\hat{p} > \frac{7}{6}2^{-\ell} \wedge \exists m . \hat{p_m} < \frac{3}{4}\hat{p}] + \Pr[\hat{p} \leq \frac{7}{6}2^{-\ell} \wedge \exists m . \hat{p_m} < \frac{3}{4}\hat{p}] \\
&\leq \Pr[\hat{p} > \frac{7}{6}2^{-\ell}] + \Pr[\exists m . \hat{p_m} < \frac{3}{4}\hat{p} | \hat{p} \leq \frac{7}{6}2^{-\ell}] \\
&\leq \Pr[\hat{p} > \frac{7}{6}2^{-\ell}] + 2n \Pr[\hat{p_m} < \frac{7}{8}2^{-\ell}] \\
&\leq e^{-n/108} + 2ne^{-n/128}
\end{aligned}
$$

Where the last two lines follow by the union bound and multiplicative Chernoff bounds.

Combining these bounds, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{ss}}_{W,S,\mathcal{C}}(k) &= \Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1] \\ &\geq 1 - e^{-n/32} + e^{-n/96} - e^{-n/108} + 2ne^{-n/128}\end{aligned}$$

The theorem follows by the definition of insecurity. □

We note that for any distribution $\mathcal{D}$ which covers $\{0,1\}^{\ell+1}$, W can easily be adapted to be a KHA-$\mathcal{D}$ attack against S; and that W can even be extended to a KDA-$U_{(\ell+1)n^2}$ attack against S by counting the occurrences of s^* for the various blocks.

6.2.3 Bidirectional communication does not help

Suppose Alice and Bob wish to communicate steganographically at a high rate, using a bidirectional channel – that is, Bob is allowed to respond to Alice's messages with messages drawn from his own channel distribution. A natural question is, "can Alice and Bob conspire to increase the rate at which Alice may securely transmit information to Bob?" We will show that an interactive stegosystem can increase the rate at which information is transmitted by at most one bit. The proof is straightforward we show that any stegosystem $\mathcal{S}$ for a bidirectional channel $\mathcal{B}$ can be converted into a stegosystem $\mathcal{S}'$ with the same rate for a unidirectional channel $\mathcal{C}$ on pairs from $\mathcal{B}.D \times \mathcal{B}.D$. $\mathcal{S}'$ runs in the same time as $\mathcal{S}$ and draws as many samples from $\mathcal{C}$ as Alice and Bob (combined) draw from $\mathcal{B}$ in each time step. Thus the upper bound $MAX_t(\mathcal{S}')$ implies that at most one bit more can be sent due to sampling, and the $MAX_{\mathcal{C}}(\mathcal{S}')$ bound implies that minimum entropy constraints still apply.

In order to address this issue, we will first need to formalize the concept of a bidirectional stegosystem.

Definition 6.3. (Bidirectional Stegosystem) A Bidirectional Stegosystem $\mathcal{S}$ for the bidirectional channel $\mathcal{B} = (D, \mathcal{H}, \mathcal{C}^A, \mathcal{C}^B)$ is a triple of algorithms:

- $\mathcal{S}$.`Encode` (abbreviated SE) takes as input a key $K \in \{0,1\}^k$, a string $m \in \{0,1\}^*$ (the *hiddentext*), a message history $h \in (D \times D)^*$, and a *session history* $T \in (D \times D)^*$.

$SE(K, I, m, h, T)$ returns a document from D, or the special symbol $\texttt{done}$

- $\mathcal{S}.\texttt{Facilitate}$ (abbreviated SF) takes as input a key K, a message history h and a session history T.

  $SF(K, h, T)$ returns a document from D.

- $\mathcal{S}.\texttt{Decode}$ (abbreviated SD) takes as input a key K, a sequence of documents $s_1||s_2||\ldots||s_l$, and a message history h.

  $SD(K, s, h)$ returns a hiddentext $m \in \{0,1\}^*$.

We denote the *transcript* of (K, m, h) by the result of setting $s_0(A) = SE(K, m, h_A, \varepsilon)$, $s_0(B) = SF(K, h_B, \varepsilon)$, and

$$s_i(A) = SE(K, m, h_A, (s_0, \ldots, s_{i-1})), s_i(B) = SF(K, h_B, (s_0, \ldots, s_{i-1}))$$

until $s_i(A) = \texttt{done}$.

We say that $\mathcal{S}$ is *correct* if for all $m \in \{0,1\}^{poly(k)}$ $\Pr[SD(K, (s_0, \ldots, s_i), h) \neq m]$ is negligible.

In addition, we formalize the notion of a bidirectional chosen hiddentext attack (BCHA) by an oracle adversary W given access to an oracle $\mathcal{O}$ drawn from one of two distributions:

- $ST_{\text{bcha}}(m, h)$: has uniformly chosen key $K \leftarrow U_k$ and responds to queries (m, h) by computing a transcript of (K, m, h).

- $CT_{\text{bcha}}(m, h)$: responds to queries by drawing a sequence from $\mathcal{B}$ of the same length as a (K, m, h) transcript.

We define BCHA advantage $(\mathbf{Adv}^{\mathsf{bcha}}_{W,\mathcal{S},\mathcal{B}}(k))$ and insecurity $(\mathbf{InSec}^{\mathsf{bcha}}_{\mathcal{S},\mathcal{B}}(t, q, \mu, k))$ according to the previous conventions.

Theorem 6.4. *Any (h, l, t)-sampling bidirectional stegosystem $\mathcal{S}$ for the bidirectional channel $\mathcal{B}$ can be converted to a $(h, l, 2t)$-sampling stegosystem $\mathcal{S}'$ for a channel $\mathcal{C}$ such that $\mathcal{R}_{\mathcal{C}}(\mathcal{S}') = \mathcal{R}_{\mathcal{B}}(\mathcal{S})$, and $\mathbf{InSec}^{\mathsf{ss}}_{\mathcal{S}',\mathcal{C}}(t', q, \mu, k) = \mathbf{InSec}^{\mathsf{bcha}}_{\mathcal{S},\mathcal{B}}(t', q, \mu, k)$.*

Proof. We define the channel $\mathcal{C}$ as a distribution on sequences of pairs from $\mathcal{B}.D \times \mathcal{B}.D$. The marginal distribution $\mathcal{C}_h$ is defined by $(\mathcal{B}.\mathcal{C}_h^A, \mathcal{B}.\mathcal{C}_h^B)$.

The stegosystem $\mathcal{S}'$ has the same keyspace as $\mathcal{S}$. The new encoding algorithm $\mathcal{S}'.SE(K, m, h)$ works by computing a transcript $s_0, \ldots, s_l$ form (K, m, h) as follows: For each i, $\mathcal{S}'.SE$ simulates $\mathcal{S}.SE$ by making draws from $\mathcal{C}_{h,s_0,\ldots,s_{i-1}}$ and extracting the first element of the pair; likewise $\mathcal{S}'.SE$ simulates $\mathcal{S}.SF$ my making draws from $\mathcal{C}_{h,s_0,\ldots,s_{i-1}}$ and extracting the second element of each pair. $\mathcal{S}'.SE(K.m, h)$ then outputs this transcript as the stegotext. Notice that since $\mathcal{S}.SE$ and $\mathcal{S}.SF$ each draw at most t samples from $\mathcal{B}.\mathcal{C}_h^A$, and $\mathcal{B}.\mathcal{C}_h^B$, respectively, $\mathcal{S}'.SE$ draws at most $2t$ samples from $\mathcal{C}_h$. The decoding algorithm $\mathcal{S}'.SD(K, s, h)$ exactly runs the decoding algorithm $\mathcal{S}.SD(K, s, h)$.

Notice that the chosen-hiddentext oracles ST, CT for $\mathcal{S}'$ are identically distributed to the bidirectional chosen-hiddentext oracles $ST_{\text{bcha}}, CT_{\text{bcha}}$ for $\mathcal{S}$. Thus any warden for $\mathcal{S}'$ is a warden for $\mathcal{S}$ and vice versa, with the advantage preserved. Thus $\mathcal{S}'$ is as secure for $\mathcal{C}$ as $\mathcal{S}$ is for $\mathcal{B}$, as claimed. It is also clear that the rate of $\mathcal{S}'$ is the same as that of $\mathcal{S}$, which completes the proof. □

6.3 Lower bound

We will prove a lower bound on the achievable rate which matches the stated upper bound, up to a multiplicative factor which converges to 1. The lower bound is proven by giving a stegosystem which achieves this rate. We will first give one that achieves rate $(1-\epsilon)MAX$ for any $\epsilon > 0$, but which has polynomial rather than negligible error rate. We will then use error-correcting codes to improve the error rate, and finally give a construction which does not require Alice or Bob to know the minimum entropy of $\mathcal{C}$.

6.3.1 With errors

We will assume for the moment that both Alice and Bob know a value l so that $\lambda < (1-\epsilon)MAX(\mathcal{C}_h^l, t)$. We let $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^\lambda$ be a pseudorandom function, and assume Alice and Bob share a key $K \in \{0,1\}^k$. The following construction allows Alice to send Bob a λ-bit message, with error probability at most $\lambda 2^{-\epsilon\lambda}$.

Construction 6.5. `OneBlock` Stegosystem

Procedure `OneBlock.Encode`:	Procedure `OneBlock.Decode`:
Input: $K \in \{0,1\}^k$, $m \in \{0,1\}^\lambda$, $h \in D^*$, N,l	**Input:** $K \in \{0,1\}^k$, $s \in D^l$, h, N
Let $i = 0$, $c[D] = 0$	**Output:** $F_K(N, h, 1, s)$
repeat:	
Draw $s_i \leftarrow \mathcal{C}_h^l$	
increment i, $c[s_i]$	
until $F_K(N, h, c[s_i], s_i) = m$ or *count* $= \lambda 2^\lambda$	
Output: s_i	

Theorem 6.6.

$$\Pr[SD(K, SE(K, m, h, N, l), h, N) \neq m] \leq e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} + \mathbf{InSec}_F^{\mathsf{prf}}(O(\lambda 2^\lambda), \lambda 2^\lambda, k)$$

Proof. We will show that when F_K is replaced by a random function f,

$$\Pr[f(SE^f(m, h, N, l)) \neq m] \leq e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} \ .$$

We can then construct a PRF adversary A with advantage at least

$$\mathbf{Adv}_{A,F}^{\mathsf{prf}}(k) \geq \Pr[SD(K, SE(K, m)) \neq m] - e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} \ ,$$

which will give the desired bound.

Let C denote the event that $\texttt{OneBlock.Encode}^f(m)$ outputs an s_i with $c[s_i] > 1$. This happens when there is at least one $j < i$ such that $s_j = s_i$. Thus by the union bound, we have

$$\Pr[\mathsf{C}] \leq \sum_{j<i} \Pr[s_j = s_i] \ .$$

Since for each j, $\Pr[s_j = s_i] \leq 2^{-H_\infty(\mathcal{C}_h^l)}$ and since $i < \lambda 2^\lambda$, we get the bound

$$\Pr[\mathsf{C}] \leq \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} \ .$$

Let D denote the event that $\texttt{OneBlock.Encode}^f(m)$ outputs $s_{\lambda 2^\lambda}$. This happens when each of the previous $\lambda 2^\lambda$ tests $f(N, h, c[s_i], s_i) = m$ fails. Since each test involves a distinct point of f, each of these happens independently with probability $1 - 1/2^\lambda$. Since the events are independent, we can bound $\Pr[\mathsf{D}]$ by

$$\Pr[\mathsf{D}] = \left(1 - \frac{1}{2^\lambda}\right)^{\lambda 2^\lambda} \leq e^{-\lambda} \ .$$

Since the only other condition under which $\mathtt{OneBlock.Encode}^f(m)$ outputs s_i is if $f(N, h, 1, s_i) = m$, we have that

$$\Pr[SD^f(SE^f(m)) \neq m] = \Pr[\mathsf{C} \wedge \mathsf{D}] \leq e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} \ .$$

We now describe a PRF adversary A for F. A^f picks $m \in \{0,1\}^\lambda$ and runs $\mathtt{OneBlock.Encode}^f(m, \varepsilon, 0, l)$ to get a sequence $s \in D^l$. A^f then outputs 1 if $f(s) \neq m$. Clearly, when A's oracle $f \leftarrow F_K$, we have

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[SD(K, SE(K, m, h, N, l), h, N) \neq m] \ ,$$

and when f is a randomly chosen function from $\{0,1\}^* \rightarrow \{0,1\}^l$, we have shown that

$$\Pr[A^f(1^k) = 1] \leq e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)}$$

It follows that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) &= \Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1] \\
&\geq \Pr[SD(K, SE(K, m, h), h) \neq m] - \left(e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)}\right)
\end{aligned}$$

And rearranging terms gives us the stated theorem:

$$\begin{aligned}
\Pr[SD(K, SE(K, m, h), h) \neq m] &\leq \mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) + e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)} \\
&\leq \mathbf{InSec}^{\mathsf{prf}}_F(O(\lambda 2^\lambda), \lambda 2^\lambda, k) + e^{-\lambda} + \lambda 2^{\lambda - H_\infty(\mathcal{C}_h^l)}
\end{aligned}$$

□

Theorem 6.7.

$$\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{OneBlock},\mathcal{C}}(t, q, q\lambda, k) \leq \mathbf{InSec}^{\mathsf{prf}}_F(t', q\lambda 2^\lambda, k) \ ,$$

Where $t' \leq t + O(q\lambda 2^\lambda)$.

Proof. Fix any nonce-respecting $W \in \mathcal{W}(t, q, q\lambda)$. We will show how to construct a PRF adversary A for F such that

$$\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) = \mathbf{Adv}^{\mathsf{ss}}_{W,\mathtt{OneBlock},\mathcal{C}}(k) \ .$$

A^f works by emulating W, responding to its queries by running $\mathtt{OneBlock.Encode}^f$; when W halts with output b, A outputs b as well. Clearly, when $f \leftarrow F_K$, we have that

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1] \ .$$

When f is a randomly chosen function, and since W is nonce-respecting, A never evaluates f on any point twice. Thus A^f is equivalent to a process which draws a new, independent function at each stage. In this model, for any $d \in D^l$, we have that $\Pr[SE(m,h) = d] = \Pr_{f,s \leftarrow \mathcal{C}_h^l}[s = d | f(s) = m]$, and since s and f are drawn independently, we have that $\Pr[SE(m,h) = d] = \Pr_{\mathcal{C}_h^l}[d]$. Thus A's responses to W's queries are distributed according to $\mathcal{C}$, so

$$\Pr[A^f(1^k) = 1] = \Pr[W^{CT}(1^k) = 1] \ .$$

Combining the cases yields:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) &= \left|\Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{ss}}_{W,\mathtt{OneBlock},\mathcal{C}}(k)
\end{aligned}$$

which proves the theorem. □

Theorem 6.8. *The rate of* `OneBlock` *is* $(1-\epsilon)MAX_{\mathcal{C}}(h, l, \lambda 2^\lambda)$.

Proof. Suppose that $MAX_{\mathcal{C}}(h, l, t) = H_\infty(\mathcal{C}_h^l)$. In this case, by choice of l and λ, `OneBlock` sends $\lambda = (1-\epsilon)H_\infty(\mathcal{C}_h^l) = (1-\epsilon)MAX$ bits in l documents. On the other hand, if $MAX_{\mathcal{C}}(h, l, \lambda 2^\lambda) = \log(\lambda 2^\lambda) = \lambda + \log \lambda$, then since `OneBlock` sends λ bits in l documents, we have that

$$\frac{\mathcal{R}_{\mathcal{C}}(\mathtt{OneBlock})}{MAX} = \frac{\lambda}{\lambda + \log \lambda} \geq (1-\epsilon) \ ,$$

where the last inequality holds for sufficiently large λ. □

6.3.2 Negligible error rate

Let $K = GF(2^\lambda)$. This next construction utilizes the following well-known fact:

Proposition 6.9. ([11]) There is a polynomial-time algorithm `Correct` to solve the following problem: given $n = 2^\lambda$ pairs $(x_1, y_1), \ldots, (x_n, y_n) \in K^2$, if there is a polynomial $p(x)$ of degree $|K| - 2t$ such that at most t pairs do not satisfy $y_i = p(x_i)$, recover p.

We will use the Berlekamp-Welch[60] algorithm, `Correct`, to reduce the probability of encoding error in the `OneBlock` construction. In the following construction, we let $n = 2^\lambda$, $\rho = \lambda 2^{-\epsilon\lambda+2}$, and $\eta = (1-2\rho)n$. The following construction securely hides messages in $\{0,1\}^{\lambda\times\eta}$:

Construction 6.10. `MultiBlock` Stegosystem

Procedure `MultiBlock.Encode`:	**Procedure** `MultiBlock.Decode`:
Input: K, $m_0, \ldots, m_{\eta-1}$, h, N	**Input:** K, $s_1, \ldots, s_n \in D^l$, h, N
for $i = 1$ to n do:	for $i = 1$ to n do:
set $x_i = \langle i \rangle$	set $x_i = \langle i \rangle$
set $y_i = \sum_{j=0}^{\eta-1} m_j x_i^j$	set $y_i =$ `OneBlock`$.SD(K, s_i, h, N)$
set $s_i =$ `OneBlock`$.SE(K, y_i, h, N, l)$	set $m_0, \ldots, m_\eta =$ `Correct`(x_i, y_i)
Output: $s_1, \ldots, s_n$	**Output:** m

Theorem 6.11.

$$\mathbf{InSec}^{\mathsf{ss}}_{\texttt{MultiBlock},\mathcal{C}}(t, q, q\eta\lambda, k) \le \mathbf{InSec}^{\mathsf{prf}}_F(t + O(q\eta\lambda 2^\lambda), q\eta\lambda 2^\lambda, k)$$

Proof. We will show how to use an arbitrary $W \in \mathcal{W}(t, q, q\eta\lambda)$ against `MultiBlock` to create an adversary $X \in \mathcal{W}(t, q\eta, q\eta\lambda)$ for `OneBlock` such that

$$\mathbf{Adv}^{\mathsf{ss}}_{W,\texttt{MultiBlock},\mathcal{C}}(k) = \mathbf{Adv}^{\mathsf{ss}}_{X,\texttt{OneBlock},\mathcal{C}}(k)\ .$$

The stated bound follows from the definition of insecurity and theorem 6.7.

$X^{\mathcal{O}}$ works by running W, and responding to W's queries using its own oracle $\mathcal{O}$. When W queries $(m_0 \cdots m_{\eta-1}, h, N)$, X produces the values $y_1, \ldots, y_n$, where $y_i = \sum_{j=0}^{\eta-1} m_j \langle i \rangle^j$, and then queries the sequences $s_i = \mathcal{O}(y_i, (h, s_1, \ldots, s_{i-1}), N)$, to produce the result $s_1, \ldots, s_n$, which is returned to W. Now when $\mathcal{O} \leftarrow ST_{\text{OneBlock}}$, it is clear that X is perfectly simulating `MultiBlock` to W, so

$$\Pr[X^{ST_{\texttt{OneBlock}}}(1^k) = 1] = \Pr[W^{ST_{\texttt{MultiBlock}}}(1^k) = 1] \ .$$

When $\mathcal{O} \leftarrow CT_{\text{OneBlock}}$, it is also clear that X is perfectly simulating samples from the channel to W, so

$$\Pr[X^{CT_{\texttt{OneBlock}}}(1^k) = 1] = \Pr[W^{CT_{\texttt{MultiBlock}}}(1^k) = 1] \ .$$

Combining these cases, we have that

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{ss}}_{X,\texttt{OneBlock},\mathcal{C}}(k) &= \left|\Pr[X^{ST_{\texttt{OneBlock}}}(1^k) = 1] - \Pr[X^{CT_{\texttt{OneBlock}}}(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST_{\texttt{MultiBlock}}}(1^k) = 1] - \Pr[W^{CT_{\texttt{MultiBlock}}}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{ss}}_{W,\texttt{MultiBlock},\mathcal{C}}(k)
\end{aligned}$$

Which completes the proof. □

Theorem 6.12. *If F is pseudorandom, then*

$$\Pr[\texttt{MultiBlock}.SD(K, \texttt{MultiBlock}.SE(K, m, h), h) \neq m] \leq e^{-n\rho/3} \ ,$$

which is negligible in $n = 2^\lambda$.

Proof. As long as there are at most ρn errors, Proposition 6.9 ensures us that `Correct` can recover the message $m_0, \ldots, m_{\eta-1}$. Thus the probability of a decoding error is at most the probability of ρn blocks having decoding error in `OneBlock.Decode`. But Theorem 6.6 states that the probability of decoding error in `OneBlock.Decode` is at most ρ when F is pseudorandom; applying a Chernoff bound yields the stated result. □

Theorem 6.13. *The rate of* `MultiBlock` *is* $(1 - \epsilon - o(1))MAX_{\mathcal{C}}(h, l, \lambda 2^\lambda)$.

Proof. The rate of `MultiBlock` is the rate of `OneBlock` multiplied by the rate of the error-correcting code used in encoding. Since this rate is $(1 - 2\rho) = 1 - \lambda 2^{-\epsilon\lambda+3}$, we have that the rate converges to 1 as $\lambda \longrightarrow \infty$, that is, the rate of the code is $(1 - o(1))$. □

6.3.3 Converging to optimal

We notice that if $\epsilon(k) = 1/\lambda$ the `MultiBlock` construction has error rate at most $e^{-\lambda 2^\lambda/3}$, and has rate $(1 - o(1))MAX_{\mathcal{C}}(h, t, l)$. Thus under appropriate parameter settings, the rate of the construction converges to the optimal rate in the limit.

6.3.4 Unknown length

Suppose Alice and Bob agree at the time of key exchange to use the `MultiBlock` stegosystem with hiddentext block size λ. Since neither Alice nor Bob necessarily know the values (α, β) such that $\mathcal{C}$ is (α, β)-informative, there is no way to calculate or exchange beforehand the stegotext block size l so that $\lambda \leq (1 - \epsilon)H_\infty(\mathcal{C}_h^l)$.

Construction 6.14. `FindBlock`

Procedure `Encode`:
Input: K, $m \in \{0,1\}^{\lambda n}$, h, N
let $l = 1$
repeat:
  let $t = F'_K(m)$
  let $s = \texttt{LEnc}(K, m\|t, l, h, N)$
  increment l
until $s \neq \perp$.
Output: s
	Procedure `Decode`:
	Input: K, $s_1, \ldots, s_t \in D^t$, h, N
	let $l = 1$
	repeat:
	  let $m\|_{\lambda n}t = \texttt{LDec}(K, s_{1\ldots(n+k)l}, h, N)$
	  increment l
	until $F'_K(m) = t$
	Output: m

Procedure `LEnc`:
Input: K, m,h, l,N
for $i = 1$ to n do:
  set $x_i = \langle i \rangle$
  set $y_i = \sum_{j=0}^{\eta-1} m_j x_i^j$
  set $s_i = \texttt{OneBlock}.SE(K, y_i, h, N, l)$
if $(\texttt{LDec}(K, s, l, h, N) \neq m)$ set $s = \perp$
Output: s
	Procedure `LDec`:
	Input: K, $s_1, \ldots, s_{n+k} \in D^l$,$l$,$h$, N
	for $i = 1$ to n do:
	  set $x_i = \langle i \rangle$
	  set $y_i = \texttt{OneBlock}.SD(K, s_i, h, N)$
	set $m_0, \ldots, m_\eta = \texttt{Correct}(x_i, y_i)$
	Output: m

The idea behind this construction is simple: Alice tries using `MultiBlock` with block lengths $l = 1, 2, \ldots$ until she finds one such that the decoding of the encoding of her message is correct. With high probability, if $H_\infty(\mathcal{C}_h^{ln}) \leq \lambda n$ decoding will fail (the block error rate will be at least $1 - \frac{1}{\lambda}$), and as we have seen, when $H_\infty(\mathcal{C}_h^{ln}) \geq (\lambda + \frac{1}{\lambda})n$ decoding fails with only negligible probability. Since $\mathcal{C}$ is (α, β)-informative, Alice will need to try at most $\lceil \frac{\alpha\lambda}{\beta} \rceil$ values of L. Alice also encodes kl bits of "check" information with her message, so that when Bob decodes with the wrong block size, he will be fooled with probability only 2^{-lk}. The rate penalty for this check data is $\frac{k}{n+k} = o(1)$ when $n = \omega(k)$. Thus for sufficiently large λ the rate of this construction will still converge to the optimal rate for $\mathcal{C}_h$.

6.4 Robust Steganography

Recall that a stegosystem is said to be substitution robust with respect to the relation R if an adversary, by making substitutions permitted by R is unable to change the decoding of a stegotext, except with negligible probability. Since an adversary is allowed to make changes to stegotexts, increasing the rate of a robust stegosystem is a more challenging task. Here we will show that if a stegosystem is robust against any δ-admissible relation R (given access to R), then it can encode at most $\log 1/\delta$ bits per document. We will also demonstrate an efficient, robust stegosystem which encodes $(1-\epsilon-o(1))\log 1/\delta$ bits per document, for any constant $\epsilon > 0$, showing that this upper bound is tight,

6.4.1 Upper Bound

Recall the definition of $\mathcal{I}(R, D) = \max_y \sum_{(x,y)\in R} \Pr_{\mathcal{D}}[x]$. We will show that any universal stegosystem for δ-admissible relations R (given access to R) which attempts to transmit more than $-\ell \log \delta$ bits in ℓ documents is either not universally secret or not universally robust.

Theorem 6.15. *Let $\mathcal{S}$ be a universal stegosystem. For every $0 < \delta < 1$, there exist a channel $\mathcal{C}$ and relation R such that*

$$\mathbf{Fail}^{R}_{\mathcal{S},\mathcal{C}}(t, 0, 0, (1+\epsilon)\ell, k) \geq 1 - 2^{-c\epsilon\sqrt{\ell}} \ ,$$

where $\mathcal{R}_{\mathcal{C}}(\mathcal{S}) = (1+\epsilon)\log(1/\delta)$, and c is a constant depending on δ.

Proof. We let $\mathcal{C}$ be the uniform distribution on n bit strings, and $R(x, y) = 1$ iff the hamming distance of x and y is at most d, where d and n are constants chosen to make $\mathcal{I}(R, \mathcal{C}) \leq \delta$. We will give an attacker W which achieves the stated success probability. For notational convenience, we define $l = -\ell \log \delta$.

W picks the challenge hiddentext $m^* \leftarrow U_l$, and gets in response the challenge stegotext $s^* \in \mathcal{S}.SE(K, m^*)$. W then uniformly picks a sequence s' subject to $|s'_i - s^*_i| \leq d$ for $1 \leq i \leq \ell$. W's output is the sequence s'.

We now compute the success probability of W. Recall that

$$\mathbf{Succ}^R_{W,\mathcal{S},\mathcal{C}}(k) = \Pr[SD(K, s') \neq m^*] \ ,$$

where this probability is taken over K, m^*, s^*, and s'. Notice that the adversary W is identical to a noisy discrete memoryless channel, with $p(s'|s^*)$ defined as the uniform distribution on $\{s \in \{0,1\}^n : |s - s^*| \leq d\}$. This channel has Shannon capacity exactly $-\log \mathcal{I}(R, \mathcal{C}) = -\log \delta$. Furthermore, any robust stegosystem is a candidate code for the channel. The strong converse to Shannon's coding theorem [62] tells us that *any* code with rate $(1 + \epsilon) \log 1/\delta$ will have average error probability at least $1 - 2^{-c\epsilon\sqrt{\ell}}$, where $c = 2^{-4n+2} \log(1/\delta)$ (which is a constant depending on δ).

Since the event that the adversary W is successful is identical to the event that a decoding error occurs in the code induced by $SE(K, \cdot), SD(K, \cdot)$, we have that

$$\mathbf{Succ}^R_{W,\mathcal{S},\mathcal{C}}(k) \geq 1 - 2^{-c\epsilon\sqrt{\ell}} \ ,$$

which satisfies the theorem. □

6.4.2 Lower Bound

In this section we will give a secure, universally δ-substitution robust stegosystem which achieves rate $(1 - \epsilon - o(1)) \log(1/\delta)$ for any $\epsilon > 0$. We will first give an exponentially time-inefficient construction with rate exactly $(1 - \epsilon) \log(1/\delta)$ and then show how to improve the computational efficiency at the expense of a $o(1)$ factor in the rate. These constructions use substantially the same ideas as were developed in chapter 5, so the analysis will also be similar.

An inefficient construction

We give a stegosystem with stegotext block size ℓ and hiddentext block size $l = (1-\epsilon)\ell \log \frac{1}{\delta}$. Suppose that the channel distribution $\mathcal{C}$ is efficiently sampleable. (Recall that $\mathcal{C}$ is efficiently sampleable if there is an efficient algorithm $\mathtt{C}$ such that, given a uniformly chosen string $s \in \{0,1\}^k$, a security parameter 1^k and history h, $\mathtt{C}(h, 1^k, s)$ is indistinguishable from $\mathcal{C}_h$). We will assume that Alice, Bob, and Ward all have access to this algorithm. Furthermore, we assume Alice and Bob share a key K to a pseudorandom function family $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$; and have a synchronized counter N.

Construction 6.16. `RLBit`

Procedure `Encode`:	**Procedure** `Decode`:
Input: K, $m \in \{0,1\}^l$, h	**Input:** K, $s_1, \ldots, s_\ell \in D^\ell$, h
for $i = 1 \ldots n$ do	set $m^* = 0$ Set $h_0 = h_1 = ()$
set $\sigma_i = \mathtt{C}(h, 1^k, F_K(N, m))$	for $m \in \{0,1\}^l$ do
increment N	for $i = 1 \ldots n$ do
set $h = (h, s_i)$	set $\sigma_i[m] = \mathtt{C}((h, h_m), 1^k, F_K(N, m))$
Output: $\sigma_1, \ldots, \sigma_n$	set $h_m = (h_m, \sigma_i[m])$
	if $(\forall 1 \leq i \leq \ell . (\sigma_i[m], s_i) \in R')$ then $m^* = m$
	set $h = (h, h_{m^*})$
	Output: m^*

The idea behind this construction is this: suppose that instead of sharing a key to a pseudorandom function F, Alice and Bob shared $(1/\delta)^l$ secret documents d_m drawn independently from $\mathcal{C}_h$. Then Alice could send Bob the message bit m by sending document d_m, and Bob could recover m by checking, for each m in turn, to see if the document he received was related (by R') to d_m. Since the adversary is R bounded and $(\mathcal{D}, R')$ is δ-admissible, the probability of a decoding error — caused either by the adversary, or by accidental draw of the d_m — would be at most $2^l \delta^\ell = \delta^{\epsilon\ell}$.

Lemma 6.17. `RLBit` is steganographically secret against a nonce-respecting chosen hiddentext attack:

$$\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{RLBit},\mathcal{C}}(t, q, ql, k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + \ell q, \ell q).$$

Proof. Let W be a passive warden which runs in time t, and makes at most q queries of total length at most ql (each query can be only l bits, because of the input type).

We construct a PRF adversary A which runs in time $t + O(q\ell)$ and makes at most $q\ell$ queries to F, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) = \mathbf{Adv}^{\mathsf{ss}}_{W,\mathtt{RLBit},\mathcal{C}}(k) \ .$$

The PRF adversary takes a function oracle f, and emulates $W(1^k)$, responding to the queries W makes to the encoder SE by using f in place of $F_K(\cdot,\cdot)$. More formally, we define the subroutine $SSE^f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ as follows:

Procedure SSE^f:
Input: $m \in \{0,1\}^l$, history h
for $i = 1 \ldots \ell$ do
    set $\sigma_i = \mathtt{C}(1^k, h, f(N, m))$
    increment N
    set $h = (h, \sigma_i)$
Output: $\sigma_1, \ldots, \sigma_\ell$

Then we define $A^f(1^k) = W^{SSE^f}(1^k)$; A's advantage over F is then:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \left|\Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[A^f(1^k) = 1]\right| \\
&= \left|\Pr[W^{ST}(1^k) = 1] - \Pr[W^{CT}(1^k) = 1]\right| \\
&= \mathbf{Adv}^{\mathsf{ss}}_{\mathtt{RLBit},\mathcal{C},W}(k) \ .
\end{aligned}$$

Where the following cases for f justify the substitutions:

- f is chosen from $F_K(\cdot,\cdot)$. Then the output of SSE^f is distributed identically to the encoding function of `RLBit`. That is,

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1] \ .$$

- f is chosen uniformly. Then by assumption on `C`, the output of SSE^f is distributed identically to samples from $\mathcal{C}_h^\ell$. that is,

$$\Pr[A^f(1^k) = 1] = \Pr[W^{CT}(1^k) = 1] \ .$$

The claim follows by the definition of insecurity. □

Lemma 6.18. $\mathbf{Fail}^{R}_{\mathtt{RLBit}}(t, q, ql, l, k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(q\ell) + 2^l\ell, q\ell + 2^l\ell, k) + \delta^{\epsilon\ell}$.

Proof. Let W be an active R-bounded (t, q, ql, l) warden. We construct a PRF adversary A which runs in time $t + O(q\ell)$, makes at most $q\ell$ PRF queries, and satisfies $\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) \geq \mathbf{Succ}^{R}_{W,\mathtt{RLBit},\mathcal{C}}(k) - \delta^{\epsilon\ell}$. A^f works by emulating W, using its function oracle f in place of $F_K(\cdot,\cdot)$ to emulate $\mathtt{RLBit.Encode}$ in responding to the queries of W. Let m_W, s_W be the hiddentext and the stegotext sequence returned by W, respectively. Then A^f returns 1 iff $SD^f(s_W, h_W) \neq m_W$. Consider the following two cases for f:

- f is chosen uniformly from all appropriate functions. Then, for each i, the stegotexts $\sigma_i = \mathtt{C}(1^k, h_i, f(N+i, m_W))$ are distributed independently according to $\mathcal{C}_{h_i}$. Consider the sequence of "alternative stegotexts" $d_i[m'] = \mathtt{C}(1^k, h'_i, f(N+i, m'))$ for each $m' \neq m_W \in \{0,1\}^l$; each of these is also distributed independently according to $\mathcal{C}_{h_i[m']}$; and since W is never given access to the $d_i[m']$, the s_i are independent of the $d_i[m']$. Now SD will fail (causing $A^f(1^k)$ to output 1) only if the event $\exists m'.\forall i.(d_i[m'], s_i) \in R'$ occurs. Because the $d_i[m']$ are independent of the actions of W, and because $(\mathcal{C}, R')$ is δ-admissible, each event $(d_i[m'], s_i) \in R'$ happens independently with probability at most δ. So for each m', the probability of failure is at most δ^ℓ, and thus by a union bound, we have that
$$\Pr[A^f(1^k) = 1] \leq \sum_{m' \in \{0,1\}^l} \delta^\ell = \delta^{\epsilon\ell} \ .$$
- f is chosen uniformly from F_K. Then $A^F(1^k)$ outputs 1 exactly when W succeeds against $\mathtt{RLBit}$, by the definition of $\mathtt{RLBit}$.
$$\Pr[A^{F_K}(1^k) = 1] = \mathbf{Succ}^{R}_{\mathtt{RLBit},W}(k) \ .$$

Taking the difference of these probabilities, we get:
$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1] \\ &= \mathbf{Succ}^{R}_{\mathtt{RLBit},W}(k) - \Pr[A^f(1^k) = 1] \\ &\geq \mathbf{Succ}^{R}_{\mathtt{RLBit},W}(l) - \delta^{\epsilon\ell} \ .\end{aligned}$$

□

Improving the run-time

Notice that because the running time of the decoding procedure for `RLBit` is exponential in ℓ, the proof of robustness is not very strong: the information-theoretic bound on the success of W is essentially polynomial in the running time of the PRF adversary we construct from W. Still, if we set $\ell = poly(\log k)$, and assume subexponential hardness for F, we obtain a negligible bound on the success probability, but a quasi-polynomial time decoding routine. We will now give a construction with a polynomial-time decoding algorithm, at the expense of a $o(1)$ factor in the rate.

As before we will assume that $\mathcal{C}$ is efficiently sampleable, that $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ is pseudorandom and both parties share a secret $K \in \{0,1\}^k$, and a synchronized counter N. As before, we will let $l = (1-\epsilon)\ell\log(1/\delta)$, but we now set ℓ so that $l = \log k$. We set an additional parameter $L = k/\log(1/\delta)$.

Construction 6.19. `RMBit`

Procedure `Encode`:
Input: K, $m_1, \ldots, m_n \in \{0,1\}^l$, h, N
for $i = 1 \ldots n+d$ do
  set $\sigma_i = \mathtt{LEnc}(K, m_{1\ldots i}, h, N, \ell)$
  set $h = (h, \sigma_i)$
set $\tau = \mathtt{LEnc}(K, m, h, N, L)$
Output: $\sigma_1, \ldots, \sigma_n, \tau$

Procedure `Decode`:
Input: K, $s_1, \ldots, s_n \in D^\ell$,$t \in D^L$, h, N
let $m^* = 0$
let $L = \mathtt{LDec}(K, s_1 \ldots s_n, \varepsilon, h, N)$
for each $m \in L$ do
  Set $\sigma_1, \ldots, \sigma_n, \tau = \mathtt{Encode}(K, m, h, N)$
  if $(\bigwedge_{1 \le i \le L} (\tau_i, t_i) \in R')$ then
    set $m^* = m$
Output: m^*

Procedure `LEnc`:
Input: K, m,h, N,len
for $i = 1 \ldots len$ do
  set $\sigma_i = \mathtt{C}(h, 1^k, F_K(N, i, m))$
  set $h = (h, \sigma_i)$
Output: $\sigma_1, \ldots, \sigma_{len}$

Procedure `LDec`:
Input: K, $s_1, \ldots, s_a$, m^*,h, N
let $L = \{\}$
for $m \in \{0,1\}^l$ do
  Let $m' = m^* \| m$
  Set $\sigma[m] = \mathtt{LEnc}(K, m', h, N, \ell)$
  if $(\bigwedge_{1 \le i \le \ell} (\sigma_i[m], s_i) \in R')$ then
    let $h[m] = (h, \sigma[m])$
    let $T = \mathtt{LDec}(K, s_{2\ldots a}, m', h[m], N)$
    set $L = L \cup \{m \| t \ : \ t \in T\}$
Output: L

The idea behind this procedure is to break a message up into l-bit blocks, and encode each one using the `RLBit.Encode` procedure; and then append the encoding of L documents of message-dependent redundancy. To decode, we iteratively attempt to match each stegotext block against each of the $2^l = k$ possible hiddentext blocks; there will always be one matching block, and with some small probability $k^{-\epsilon}$ there will be an additional match. We perform a depth-first search on these matches to find a list of candidate messages, and then test each message to see whether the redundant encoding matches. Any candidate match from the depth-first search will also have matching redundancy with probability 2^{-k}, and a union bound will thus bound the probability of a decoding failure by $(1 + \frac{1}{\epsilon})2^{-k}$. Furthermore, the total expected number of nodes explored by `Decode` is at most $(1+\frac{1}{\epsilon})n$; thus our reduction will be efficient.

Theorem 6.20. `RMBit` *is steganographically secret against a nonce-respecting chosen hiddentext attack:*

$$\mathbf{InSec}^{\mathsf{ss}}_{\mathtt{RMBit},\mathcal{C}}(t, q, l\mu, k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t + O(\mu\ell), \mu\ell).$$

Proof. Let W be a passive warden which runs in time t, and makes at most q queries of total length at most $l\mu$ (each query must be a multiple of l bits, because of the input type). We construct a PRF adversary A which runs in time $t + O(\mu\ell)$ and makes at most $\mu\ell$ queries to F, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) = \mathbf{Adv}^{\mathsf{ss}}_{W,\mathtt{RMBit},\mathcal{C}}(k) \ .$$

The PRF adversary takes a function oracle f, and emulates $W(1^k)$, responding to the queries W makes to its oracle $\mathcal{O}$ by running `RMBit.Encode`, using f in place of $F_K(\cdot,\cdot)$. Consider the following cases for f:

- f is chosen from $F_K(\cdot,\cdot)$. Then the responses to W's queries are distributed identically to the encoding function of `RMBit`. That is,

$$\Pr[A^{F_K}(1^k) = 1] = \Pr[W^{ST}(1^k) = 1] \ .$$

- f is chosen uniformly. Then by assumption on `C`, the response to each query by W is distributed identically to samples from $\mathcal{C}_h^{\ell}$. that is,

$$\Pr[A^{f}(1^k) = 1] = \Pr[W^{CT}(1^k) = 1] \ .$$

A's advantage over F is then:

$$\begin{aligned}\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \left|\Pr[A^{F_K}(1^k)=1]-\Pr[A^f(1^k)=1]\right| \\ &= \left|\Pr[W^{ST}(1^k)=1]-\Pr[A^f(1^k)=1]\right| \\ &= \left|\Pr[W^{ST}(1^k)=1]-\Pr[W^{CT}(1^k)=1]\right| \\ &= \mathbf{Adv}^{\mathsf{ss}}_{W,\mathtt{RMBit},\mathcal{C}}(k)\ .\end{aligned}$$

The claim follows by the definition of insecurity. □

Theorem 6.21. `RMBit` *is robust:*

$$\mathbf{Fail}^{R}_{\mathtt{RMBit}}(t,q,l\mu,ln,k) \leq \mathbf{InSec}^{\mathsf{prf}}_{F}(t',2n(1+1/\epsilon)+l(\mu+n),k)+(1+1/\epsilon)2^{-k}+(e/4)^n\ ,$$

where $t' \leq t+O((l+\mu)n)+O((1+1/\epsilon)kn)$.

Proof. Let W be an active R-bounded $(t,q,l\mu,ln)$ warden. We construct a PRF adversary A which runs in time t', makes at most $2n(1+1/\epsilon)+l(\mu+n)$ PRF queries, and satisfies $\mathbf{Adv}^{\mathsf{prf}}_{A,F}(k) \geq \mathbf{Succ}^{R}_{W,\mathtt{RMBit},\mathcal{C}}(k)-(1+1/\epsilon)2^{-k}-(e/4)^n$. A^f works by emulating W, using its function oracle f in place of $F_K(\cdot,\cdot)$ to emulate `RMBit.Encode` in responding to the queries of W. Let m^*,s^* be the hiddentext and the stegotext sequence returned by W, respectively. Then A^f returns 1 iff $SD^f(s^*,h^*) \neq m^*$. To ensure that the number of queries and running time are at most t', and $2n(1+1/\epsilon)+l(\mu+n)$, we halt whenever SD^f queries makes more than $2n(1+1/\epsilon)$ to f, an event we will denote by TB. We will show that $\Pr[\mathsf{TB}] \leq (e/4)^n$ when f is a randomly chosen function. Thus we can neglect this case in our analyses of the cases for f.

Consider the following two cases for f:

- f is chosen uniformly from all appropriate functions. Then, a decoding error happens when there exists another $m \in \{0,1\}^{ln}$ such that for all (i,j), $1 \leq i \leq \ell$, $1 \leq j \leq n$, we have $(s_{(j-1)n+i}, \mathtt{LEnc}^f(m_{1\ldots j})_i) \in R$; and also $(s_{\ell n+i}, \mathtt{LEnc}^f(m)_i) \in R$ for all i, $1 \leq i \leq L$. Let j be the least j such that $m_j \neq m^*_j$. Then for blocks $m_{j+1},\ldots,m_n$, the ℓ-document blocks $\mathtt{LEnc}^f(m_{1\ldots j+i})$ are independent of σ^*_{j+i}. Thus for such m, the probability of a match is at most $\delta^{\ell(n-j)+L} = 2^{-k}\delta^{(n-j)\ell}$.

Since there are $2^{l(n-j)}$ messages matching m^* in the first j blocks, we have that

$$\begin{aligned}
\Pr[A^f(1^k) = 1] &= \Pr[SD^f(s^*) \neq m^*] \\
&\leq \Pr[\exists m \neq m^*. \bigwedge_{1 \leq i \leq \ell n + L} (s_i(m_{1\ldots i/l}), s_i^*) \in R] \\
&\leq \sum_{j=0}^{n} 2^{l(n-j)} 2^{-k} \delta^{(n-j)\ell} \\
&\leq 2^{-k} \sum_{j=0}^{\infty} \delta^{\epsilon \ell j} \\
&= 2^{-k} \frac{1}{1 - \delta^{\epsilon \ell}} \\
&\leq 2^{-k}(1 + 1/\epsilon)
\end{aligned}$$

- f is chosen uniformly from F_K. Then $A^F(1^k)$ outputs 1 exactly when W succeeds against `RMBit`, by the definition of `RMBit`.

$$\Pr[A^{F_K}(1^k) = 1] = \mathbf{Succ}^{R}_{W,\texttt{RMBit},\mathcal{C}}(k) \ .$$

Taking the difference of these probabilities, we get:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{prf}}_{F,A}(k) &= \Pr[A^{F_K}(1^k) = 1] - \Pr[A^f(1^k) = 1] \\
&= \mathbf{Succ}^{R}_{\texttt{RMBit},W}(k) - \Pr[A^f(1^k) = 1] \\
&\geq \mathbf{Succ}^{R}_{\texttt{RMBit},W}(l) - (1 + 1/\epsilon)2^{-k} - \Pr[\mathsf{TB}] \ .
\end{aligned}$$

It remains to show that $\Pr[\mathsf{TB}] \leq (e/4)^n$. Notice that the expected number of queries to f by A is just the number of messages that match a $j\ell$-document prefix of s^*, for $1 \leq j \leq n$, times k. Let $X_m = 1$ if $m \in \{0,1\}^{j\ell}$ matches a j-block prefix of s^*. Let $X = \sum_{j=1}^{n} \sum_{m \in \{0,1\}^{j\ell}} X_m$ denote the number of matching prefix messages. Then $n \leq E[X] \leq n(1 + 1/\epsilon)$, and a Chernoff bound gives us

$$\begin{aligned}
\Pr[X > 2n(1 + 1/\epsilon)] &\leq \Pr[X > 2E[X]] \\
&\leq (e/4)^{E[X]} \\
&\leq (e/4)^n
\end{aligned}$$

which completes the proof. $\square$

Theorem 6.22. $\mathcal{R}_{\mathcal{C}}(\texttt{RMBit}) = (1-\epsilon)\log(1/\delta) - o(1)$

Proof. For a message of length $ln = (1-\epsilon)\log(1/\delta)\ell n$, *RMBit* transmits $\ell n + L = \ell n + k/\log(1/\delta)$ documents. Thus the rate is

$$\begin{aligned}\frac{(1-\epsilon)\log(1/\delta)\ell n}{\ell n + k/\log(1/\delta)} &= (1-\epsilon)\log(1/\delta) - \frac{O(k)}{\ell n + O(k)} \\ &\geq (1-\epsilon)\log(1/\delta) - \frac{k}{n}\end{aligned}$$

For any choice of $n = \omega(k)$, the second term is $o(1)$, as claimed. $\square$

Chapter 7

Covert Computation

7.1 Introduction

Secure two-party computation allows Alice and Bob to evaluate a function of their secret inputs so that neither learns anything other than the output of the function. A real-world example that is often used to illustrate the applications of this primitive is when Alice and Bob wish to determine if they are romantically interested in each other. Secure two-party computation allows them to do so without revealing their true feelings *unless they are both attracted.* By securely evaluating the AND of the bits representing whether each is attracted to the other, both parties can learn if there is a match without risking embarrassment: if Bob is not interested in Alice, for instance, the protocol does not reveal whether Alice is interested in him. So goes the example.

However, though often used to illustrate the concept, this example is not entirely logical. The very use of two-party computation already reveals possible interest from one party: "would you like to determine if we are both attracted to each other?"

A similar limitation occurs in a variety of other applications where the very use of the primitive raises enough suspicion to defeat its purpose. To overcome this limitation we introduce *covert two-party computation*, which guarantees the following (in addition to leaking no additional knowledge about the individual inputs): (A) no outside eavesdropper can determine whether the two parties are performing the computation or simply communicating as they normally do; (B) before learning $f(x_A, x_B)$, neither party can tell whether the other is running the protocol; (C) at any point prior to or after the conclusion of the protocol, each party can only determine if the other ran the protocol insofar as they can distinguish $f(x_A, x_B)$ from uniformly chosen random bits. By defining a functionality $g(x_A, x_B)$ such that $g(x_A, x_B) = f(x_A, x_B)$ whenever $f(x_A, x_B) \in Y$ and $g(x_A, x_B)$ is pseudorandom otherwise, covert two-party computation allows the construction of protocols that return $f(x_A, x_B)$ only when it is in a certain set of interesting values Y but for which *neither party can determine whether the other even ran the protocol whenever* $f(x_A, x_B) \notin Y$. Among the many important potential applications of covert two-party computation we mention the following:

- **Dating.** As hinted above, covert two-party computation can be used to properly determine if two people are romantically interested in each other. It allows a person to approach another and perform a computation hidden in their normal-looking messages such that: (1) if both are romantically interested in each other, they *both* find out; (2) if none or only one of them is interested in the other, neither will be able to determine that a computation even took place. In case both parties are romantically interested in each other, it is important to guarantee that *both* obtain the result. If one of the parties can get the result while ensuring that the other one doesn't, this party would be able to learn the other's input by pretending he is romantically interested; there would be no harm for him in doing so since the other would never see the result. However, if the protocol is fair (either both obtain the result or neither of them does), parties have a deterrence from lying.

- **Cheating in card games.** Suppose two parties playing a card game want to determine whether they should cheat. Each of them is self-interested, so cheating should not occur unless both players can benefit from it. Using covert two-party computation with both players' hands as input allows them to compute if they have an opportunity to benefit from cheating while guaranteeing that: (1) neither player finds out whether the other attempted to cheat unless they can both benefit from it; (2) none of the other players can determine if the two are secretly planning to collude.

- **Bribes.** Deciding whether to bribe an official can be a difficult problem. If the official is corrupt, bribery can be extremely helpful and sometimes necessary. However, if the official abides by the law, attempting to bribe him can have extremely negative consequences. Covert two-party computation allows individuals to approach officials and negotiate a bribe with the following guarantees: (1) if the official is willing to accept bribes and the individual is willing to give them, the bribe is agreed to; (2) if at least one of them is not willing to participate in the bribe, neither of them will be able to determine if the other attempted or understood the attempt of bribery; (3) the official's supervisor, even after seeing the entire sequence of messages exchanged, will not be able to determine if the parties performed or attempted bribery.

- **Covert Authentication.** Imagine that Alex works for the CIA and Bob works for Mossad. Both have infiltrated a single terrorist cell. If they can discover their "mutual interest" they could pool their efforts; thus both should be looking for potential collaborators. On the other hand, suggesting something out of the ordinary is happening to a normal member of the cell would likely be fatal. Running a covert computation in which both parties' inputs are their (unforgeable) credentials and the result is 1^k if they are allies and uniform bits otherwise will allow Alex and Bob to authenticate each other such that if Bob is NOT an ally, he will not know that Alex was even asking for authentication, and vice-versa. (Similar situations occur in, e.g., planning a *coup d'etat* or constructing a zombie network)

- **Cooperation between competitors.** Imagine that Alice and Bob are competing online retailers and both are being compromised by a sophisticated cracker. Because of the volume of their logs, neither Alice nor Bob can draw a reliable inference about the location of the hacker; statistical analysis indicates about twice as many attack events are required to isolate the cracker. Thus if Alice and Bob were to compare their logs, they could solve their problem. But if Alice admits she is being hacked and Bob is not, he will certainly use this information to take her customers; and vice-versa. Using covert computation to perform the log analysis online can break this impasse. If Alice is concerned that Bob might fabricate data to try and learn something from her logs, the computation could be modified so that when an attacker is identified, the output is both an attacker and a signed contract stating that Alice is due a prohibitively large fine (for instance, $1 Billion US) if she can determine that Bob falsified his log, and vice-versa. Similar situations occur whenever cooperation might benefit mutually distrustful competitors.

Our protocols make use of provably secure steganography [4, 7, 34, 53] to hide the computation in innocent-looking communications. Steganography alone, however, is not enough. Combining steganography with two-party computation in the obvious black-box manner (i.e., forcing all the parties participating in an ordinary two-party protocol to communicate steganographically) yields protocols that are undetectable to an outside observer but does not guarantee that the participants will fail to determine if the computation took place. Depending on the output of the function, we wish to hide that the computation took place even from the participants themselves.

Synchronization, and who knows what?

Given the guarantees that covert-two party computation offers, it is important to clarify what the parties know and what they don't. We assume that both parties know a common circuit for the function that they wish to evaluate, that they know which role they will play in the evaluation, and that they know when to start evaluating the circuit if the computation is going to occur. An example of such "synchronization" information could be: "if we will determine whether we both like each other, the computation will start with the first message exchanged after 5pm." (Notice that since such details can be published as part of the protocol specification, there is no need for either party to indicate that they wish to compute anything at all) We assume adversarial parties know all such details of the protocols we construct.

Hiding Computation vs. Hiding inputs

Notice that covert computation is not about hiding *which function* Alice and Bob are interested in computing, which could be accomplished via standard SFE techniques: *Covert Computation hides the fact that Alice and Bob are interested in computing a function at all.* This point is vital in the case of, e.g., covert authentication, where expressing a desire to do *anything* out of the ordinary could result in the death of one of the parties. In fact, we assume that the specific function to be computed (if any) is known to all parties. This is analogous to the difference in security goals between steganography – where the adversary is assumed to know which message, if any, is hidden – and encryption, where the adversary is trying to decide which of two messages are hidden.

Roadmap.

The high-level view of our presentation is as follows. First, we will define the security properties of covert two-party computation. Then we will present two protocols. The first protocol we present will be a modification of Yao's "garbled circuit" two-party protocol in which, except for the oblivious transfer, all messages generated are indistinguishable from uniform random bits. We construct a protocol for oblivious transfer that generates messages that are indistinguishable from uniform random bits (under the Decisional Diffie-Hellman assumption) to yield a complete protocol for two-party secure function evaluation that generates messages indistinguishable from random bits. We then use steganography to transform this into a protocol that generates messages indistinguishable from "ordinary" communications. The protocol thus constructed, however, is not secure against malicious adversaries nor is it fair (since neither is Yao's protocol by itself). We therefore construct another protocol, which uses our modification of Yao's protocol as a subroutine, that satisfies fairness and is secure against malicious adversaries, in the Random Oracle Model. The major difficulty in doing so is that the standard zero-knowledge-based techniques for converting a protocol in the honest-but-curious model into a protocol secure against malicious adversaries cannot be applied in our case, since they reveal that that the other party is running the protocol.

Related Work.

Secure two-party computation was introduced by Yao [63]. Since then, there have been several papers on the topic and we refer the reader to a survey by Goldreich [26] for further references. Constructions that yield fairness for two-party computation were introduced by Yao [64], Galil et al. [24], Brickell et al. [15], and many others (see [51] for a more complete list of such references). The notion of covert two-party computation, however, appears to be completely new.

Notation.

We say a function $\mu : \mathbb{N} \to [0,1]$ is *negligible* if for every $c > 0$, for all sufficiently large k, $\mu(k) < 1/k^c$. We denote the length (in bits) of a string or integer s by $|s|$ and the concatenation of string s_1 and string s_2 by $s_1||s_2$. We let U_k denote the uniform distribution on k bit strings. If $\mathcal{D}$ is a distribution with finite support X, we define the *minimum entropy* of $\mathcal{D}$ as $H_\infty(\mathcal{D}) = \min_{x\in X}\{\log_2(1/\Pr_{\mathcal{D}}[x])\}$. The *statistical distance* between two distributions $\mathcal{C}$ and $\mathcal{D}$ with joint support X is defined by $\Delta(\mathcal{C},\mathcal{D}) = (1/2)\sum_{x\in X} |\Pr_{\mathcal{D}}[x] - \Pr_{\mathcal{C}}[x]|$. Two sequences of distributions, $\{\mathcal{C}_k\}_k$ and $\{\mathcal{D}_k\}_k$, are called *computationally indistinguishable*, written $\mathcal{C} \approx \mathcal{D}$, if for any probabilistic polynomial-time $\mathbf{A}$, $\mathbf{Adv}^{\mathcal{C},\mathcal{D}}_{\mathbf{A}}(k) = |\Pr[\mathbf{A}(\mathcal{C}_k) = 1] - \Pr[\mathbf{A}(\mathcal{D}_k) = 1]|$ is negligible in k.

7.2 Covert Two-Party Computation Against Semi-Honest Adversaries

We now present a protocol for covert two-party computation that is secure against semi-honest adversaries in the standard model (without Random Oracles) and assumes that the decisional Diffie-Hellman problem is hard. The protocol is based on Yao's well-known function evaluation protocol [63].

We first define covert two-party computation formally, following standard definitions for secure two-party computation, and we then describe Yao's protocol and the necessary modifications to turn it into a covert computation protocol. The definition presented in this section is only against honest-but-curious adversaries and is unfair in that only one of the parties obtains the result. In Section 4 we will define covert two-party computation against malicious adversaries and present a protocol that is fair: either both parties obtain the result or neither of them does. The protocol in Section 4 uses the honest-but-curious protocol presented in this section as a subroutine.

7.2.1 Definitions

Formally, a two-party, n-round protocol is a pair $\Pi = (P_0, P_1)$ of programs. The computation of Π proceeds as follows: at each round, P_0 is run on its input x_0, the security parameter 1^k, a state s_0, and the (initially empty) history of messages exchanged so far, to produce a new message m and an internal state s_0. The message m is sent to P_1, which is run on its input x_1, the security parameter 1^k, a state s_1, and the history of messages exchanged so far to produce a message that is sent back to P_0, and a state s_1 to be used in the next round. Denote by $\langle P_0(x_0), P_1(x_1)\rangle$ the *transcript* of the interaction of P_0 with input x_0 and P_1 with input x_1. This transcript includes all messages exchanged between P_0 and P_1 along with the timestep in which they were sent. After n rounds, each party $P \in \{P_0, P_1\}$ halts with an output, denoted by $\Pi_P(x_0, x_1) = \Pi_P(\bar{x})$. We say that Π *correctly realizes the functionality* f if for at least one $P \in \{P_0, P_1\}$, $\Pr[\Pi_P(\bar{x}) = f(\bar{x})] \geq 1 - \nu(k)$, where ν is negligible.

For $\sigma \in \{0, 1\}$, we denote by $V_\Pi^{P_\sigma}(x_0, x_1)$ the *view* of party P_σ on input x_σ when interacting with $P_{1-\sigma}$ on input $x_{1-\sigma}$. The view includes P_σ's input x_σ, private random bits, and all messages sent by P_0 and P_1. We say Π *securely realizes the functionality* f if Π correctly realizes f and, for any P'_σ and $x_{1-\sigma}$, there is a *simulator* P''_σ and an x_σ such that $P''_\sigma(f(x_0, x_1)) \approx V_\Pi^{P'_\sigma}(x_0, x_1)$. Notice that given $f(x_0, x_1)$, P'_σ could just use P''_σ to simulate his interaction with $P_{1-\sigma}$ without actually running Π. Thus if Π securely implements f, neither party learns more from the interaction than could be learned from just $f(x_0, x_1)$.

Define the view of party P interacting in protocol Π up through round j by $V_{\Pi,j}^P(\bar{x})$. When party P_σ is not executing Π but is drawing from $\mathcal{B}$ instead, we denote this "protocol" by $\Pi : \mathcal{B}_\sigma$.

Definition 7.1. (Covert two-party protocol against honest-but-curious adversaries) We say an n-round, two-party protocol (P_0, P_1) *covertly realizes the functionality f for bidirectional channel $\mathcal{B}$* if it securely realizes f and if it has the following additional properties:

1. (External covertness): For any input $\bar{x}$, $\langle P_0(x_0), P_1(x_1)\rangle \approx \mathcal{B}$.
2. (Internal covertness): For any input $\bar{x}$, $V_{\Pi,n}^{P_0}(\bar{x}) \approx V_{\Pi:\mathcal{B}_1,n}^{P_0}(\bar{x})$ and $V_{\Pi,n-1}^{P_1}(\bar{x}) \approx V_{\Pi:\mathcal{B}_0,n-1}^{P_1}(\bar{x})$.
3. (Final Covertness): For every PPT D there exists a PPT D' and a negligible ν such that for any x_1 and any distribution X_0, $\mathbf{Adv}_D^{V_\Pi^{P_1}(X_0,x_1),V_{\Pi:\mathcal{B}_0}^{P_1}(X_0,x_1)}(k) \leq \mathbf{Adv}_{D'}^{f(X_0,x_1),U_l}(k) + \nu(k)$.

In other words, until the final round, neither party can distinguish between the case that the other is running the protocol or just drawing from $\mathcal{B}$; and after the final message, P_0 still cannot tell, while P_1 can only distinguish the cases if $f(x_0, x_1)$ and U_m are distinguishable. Note that property 2 implies property 1, since P_0 could apply the distinguisher to his view (less the random bits).

We will slightly abuse notation and say that a protocol which has messages indistinguishable from random bits (even given one party's view) is *covert for the uniform channel $\mathcal{U}$.*

7.2.2 Yao's Protocol For Two-Party Secure Function Evaluation

Yao's protocol [63] securely (not covertly) realizes any functionality f that is expressed as a combinatorial circuit. Our description is based on [46]. The protocol is run between two parties, the *Input Owner A* and the *Program Owner B*. The input of A is a value x, and the input of B is a description of a function f. At the end of the protocol, B learns $f(x)$ (and nothing else about x), and A learns nothing about

f. The protocol requires two cryptographic primitives, pseudorandom functions and oblivious transfer, which we describe here for completeness.

Pseudorandom Functions.

Let $\{F : \{0,1\}^k \times \{0,1\}^{L(k)} \to \{0,1\}^{l(k)}\}_k$ denote a sequence of function families. Let $\mathbf{A}$ be an oracle probabilistic adversary. We define the *prf-advantage of* $\mathbf{A}$ *over* F as $\mathbf{Adv}^{\mathsf{prf}}_{\mathbf{A},F}(k) = |\Pr_K[\mathbf{A}^{F_K(\cdot)}(1^k) = 1] - \Pr_g[\mathbf{A}^g(1^k) = 1]|$, where $K \leftarrow U_k$ and g is a uniformly chosen function from $L(k)$ bits to $l(k)$ bits. Then F is *pseudorandom* if $\mathbf{Adv}^{\mathsf{prf}}_{\mathbf{A},F}(k)$ is negligible in k for all polynomial-time $\mathbf{A}$. We will write $F_K(\cdot)$ as shorthand for $F_{|K|}(K, \cdot)$ when $|K|$ is known.

Oblivious Transfer.

1-out-of-2 oblivious transfer (OT^2_1) allows two parties, the *sender* who knows the values m_0 and m_1, and the *chooser* whose input is $\sigma \in \{0,1\}$, to communicate in such a way that at the end of the protocol the chooser learns m_σ, while learning nothing about $m_{1-\sigma}$, and the sender learns nothing about σ. Formally, let $\mathcal{O} = (S, C)$ be a pair of interactive PPT programs. We say that $\mathcal{O}$ is *correct* if $\Pr[\mathcal{O}_C((m_0, m_1), \sigma) = m_\sigma] \geq 1 - \epsilon(k)$ for negligible ϵ. We say that $\mathcal{O}$ has *chooser privacy* if for any PPT S' and any m_0, m_1, $\left|\Pr[S'(\langle S'(m_0, m_1), C(\sigma)\rangle) = \sigma] - \frac{1}{2}\right| \leq \epsilon(k)$ and $\mathcal{O}$ has *sender privacy* if for any PPT C' there exists a σ and a PPT C'' such that $C''(m_\sigma) \approx V^{C'}_{\Pi}((m_0, m_1), \sigma)$. We say that $\mathcal{O}$ *securely realizes the functionality* OT^2_1 if $\mathcal{O}$ is correct and has chooser and sender privacy.

Yao's Protocol.

Yao's protocol is based on expressing f as a combinatorial circuit. Starting with the circuit, the program owner B assigns to each wire i two random k-bit values (W_i^0, W_i^1) corresponding to the 0 and 1 values of the wire. It also assigns a random permutation π_i over $\{0,1\}$ to the wire. If a wire has value b_i we say it has "garbled" value $(W_i^{b_i}, \pi_i(b_i))$. To each gate g, B assigns a unique identifier I_g and a table T_g which enables computation of the garbled output of the gate given the garbled inputs.

Given the garbled inputs to g, T_g does not disclose any information about the garbled output of g for any other inputs, nor does it reveal the actual values of the input bits or the output bit.

Assume g has two input wires (i, j) and one output wire *out* (gates with higher fan in or fan out can be accommodated with straightforward modifications). The construction of T_g uses a pseudorandom function F whose output length is $k + 1$. The table T_g is as follows:

$\pi_i(b_i)$	$\pi_j(b_j)$	value
0	0	$(W_{out}^{g(b_i,b_j)}, \pi_o(b_{out})) \oplus F_{W_j^{b_j}}(I_g, 0) \oplus F_{W_i^{b_i}}(I_g, 0)$
0	1	$(W_{out}^{g(b_i,b_j)}, \pi_o(b_{out})) \oplus F_{W_j^{b_j}}(I_g, 0) \oplus F_{W_i^{b_i}}(I_g, 1)$
1	0	$(W_{out}^{g(b_i,b_j)}, \pi_o(b_{out})) \oplus F_{W_j^{b_j}}(I_g, 1) \oplus F_{W_i^{b_i}}(I_g, 0)$
1	1	$(W_{out}^{g(b_i,b_j)}, \pi_o(b_{out})) \oplus F_{W_j^{b_j}}(I_g, 1) \oplus F_{W_i^{b_i}}(I_g, 1)$

To compute $f(x)$, B computes garbled tables T_g for each gate g, and sends the tables to A. Then, for each circuit input wire i, A and B perform an oblivious transfer, where A plays the role of the chooser (with $\sigma = b_i$) and B plays the role of the sender, with $m_0 = W_i^0 \| \pi_i(0)$ and $m_1 = W_i^1 \| \pi_i(1)$. A computes $\pi_j(b_j)$ for each output wire j of the circuit (by trickling down the garbled inputs using the garbled tables) and sends these values to B, who applies π_j^{-1} to learn b_j. Alternatively, B can send the values π_j (for each circuit output wire j) to A, who then learns the result. Notice that the first two columns of T_g can be implicitly represented, leaving a "table" which is indistinguishable from uniformly chosen bits.

7.2.3 Steganographic Encoding

We use provably secure steganography to transform Yao's protocol into a covert two-party protocol; we also use it as a building block for all other covert protocols presented in this paper. For completeness we state a construction that has appeared in various forms in [4, 16, 34]. Let $\mathcal{HASH}$ denote a family of hash functions $H : D \to \{0,1\}^c$ which is *pairwise independent*, that is, for any $x_1 \neq x_2 \in D$, for any $y_1, y_2 \in \{0,1\}^m$, $\Pr_H[H(x_1) = y_1 \wedge H(x_2) = y_2] = 1/2^{2m}$. Let $\mathcal{D}$ denote an arbitrary probability distribution on D satisfying $H_\infty(\mathcal{D}) = \ell(k)$ where k is the security parameter. The following constructions hide and recover m uniformly-chosen bits in a distribution indistinguishable from $\mathcal{D}$ when $\ell(k) - m = \omega(\log k)$ and $m = O(\log k)$.

Construction 7.2. (Basic steganographic encoding/decoding routines)

Procedure $\texttt{Basic\_Encode}^{\mathcal{D}}$:	**Procedure** $\texttt{Basic\_Decode}$:
Input: $H \in \mathcal{HASH}$, $c \in \{0,1\}^m$	**Input:** $H \in \mathcal{HASH}$, $s \in D$
Let $j = 0$	set $c = H(s)$
repeat:	**Output:** c
sample $s \leftarrow \mathcal{D}$, increment j	
until $H(s) = c$ OR $(j > k)$	
Output: s	

Proposition 7.3. Let $H \leftarrow \mathcal{HASH}$. Then

$$\Delta\left((H, \texttt{Basic\_Encode}^{\mathcal{D}}(H, U_m)), (H, \mathcal{D})\right) \leq 2^{-(\ell(k)-m)/2+1} .$$

The result follows from the Leftover Hash Lemma ([33], Lemma 4.8). Intuitively, it guarantees that $\texttt{Basic\_Encode}(c)$ will be (statistically) indistinguishable from the messages exchanged in a bidirectional channel whenever c is a uniformly chosen bit string. (When we refer to $\texttt{Basic\_Encode}$ with only a single argument, we implicitly assume that an appropriate h has been chosen and is publicly accessible to all parties.)

Thus, to guarantee covertness for channel $\mathcal{B}$, we will ensure that all our protocols generate messages that are indistinguishable from uniformly chosen random bits and then encode these messages with $\texttt{Basic\_Encode}$. Formally, suppose $\Pi = (P_0, P_1)$ is an arbitrary two-party protocol which securely realizes the functionality f. We will construct a protocol $\Sigma^\Pi = (S_0^{P_0}, S_1^{P_1})$ which has the property that if $V_\Pi^{P_b}(\bar{x})$ is indistinguishable from uniformly chosen bits (that is, Π covertly realizes f for the uniform channel), then Σ^Π covertly realizes the functionality f for channel $\mathcal{B}$. We assume that P_0, P_1 have the property that, given a partial input, they return the string ε, indicating that more bits of input are needed. Then $S_b^{P_b}$ has the following round function (which simply uses $\texttt{Basic\_Encode}$ and $\texttt{Basic\_Decode}$ to encode and decode all messages exchanged by P_0 and P_1):

Construction 7.4. (Transformation to a covert protocol)

Procedure $S_b^{P_b}$:

Input: history $h \in \mathcal{H}$, $\mathtt{state}$, document $s \in D$

draw $d \leftarrow \mathcal{B}_h^{P_b}$

if ($\mathtt{state}$.status = "receiving") then

set $\mathtt{state}.msg = \mathtt{state}.msg \| \mathtt{Basic\_Decode}(s)$

set $c = P_b(\mathtt{state}.msg)$

if $(c \neq \varepsilon)$ set $\mathtt{state}$.status = "sending"; set $\mathtt{state}.msg = c$

if ($\mathtt{state}$.status = "sending") then

if $(d \neq \perp)$ then

set $c =$ first m bits of $\mathtt{state}.msg$

set $\mathtt{state}.msg = \mathtt{state}.msg$ without the first m bits

set $d = \mathtt{Basic\_Encode}^{(\mathcal{C}_h^{P_b} \neq \perp)}(c)$

if $\mathtt{state}.msg =$ "" set $\mathtt{state}$.status = "receiving"

Output: message d, $\mathtt{state}$

Theorem 7.5. *If* Π *covertly realizes the functionality* f *for the uniform channel, then* Σ^Π *covertly realizes* f *for the bidirectional channel* $\mathcal{B}$.

Proof. Let k^c be an upper bound on the number of bits in $\langle P_0(x_0), P_1(x_1)\rangle$. Then Σ^Π transmits at most $2k^c/m$ (non-empty) documents. Suppose there is a distinguisher D for $V_\Sigma^{S_b}(\bar{x})$ from $V_{\Sigma:\mathcal{B}_{1-b}}^{S_b}(\bar{x})$ with significant advantage ϵ. Then D can be used to distinguish $V_\Pi^{P_b}(\bar{x})$ from $V_{\Pi:\mathcal{U}_{1-b}}^{P_b}(\bar{x})$, by simulating each round as in Σ to produce a transcript T; If the input is uniform, then $\Delta(T, \mathcal{B}) \leq (k^c/m)2^{2-(\ell(k)-m)/2} = \nu(k)$, and if the input is from Π, then T is identical to $V_\Sigma^{S_b}(\bar{x})$. Thus D's advantage in distinguishing Π from $\Pi : \mathcal{U}_{1-b}$ is at least $\epsilon - \nu(k)$. $\square$

IMPORTANT: For the remainder of the paper we will present protocols Π that covertly realize f for $\mathcal{U}$. It is to be understood that the final protocol is meant to be Σ^Π, and that when we state that "Π covertly realizes the functionality f" we are referring to Σ^Π.

7.2.4 Covert Oblivious Transfer

As mentioned above, we guarantee the security of our protocols by ensuring that all the messages exchanged are indistinguishable from uniformly chosen random bits. To this effect, we present a modification of the Naor-Pinkas [45] protocol for oblivious transfer that ensures that all messages exchanged are indistinguishable from uniform when the input messages m_0 and m_1 are uniformly chosen. Our protocol relies on the well-known integer decisional Diffie-Hellman assumption:

Integer Decisional Diffie-Hellman.

Let P and Q be primes such that Q divides $P-1$, let $\mathbb{Z}_P^*$ be the multiplicative group of integers modulo P, and let $g \in \mathbb{Z}_P^*$ have order Q. Let $\mathbf{A}$ be an adversary that takes as input three elements of $\mathbb{Z}_P^*$ and outputs a single bit. Define the *DDH advantage of* $\mathbf{A}$ *over* (g, P, Q) as: $\mathbf{Adv}_{\mathbf{A}}^{\mathsf{ddh}}(g, P, Q) = |\Pr_{a,b,r}[\mathbf{A}_r(g^a, g^b, g^{ab}, g, P, Q) = 1] - \Pr_{a,b,c,r}[\mathbf{A}_r(g^a, g^b, g^c, g, P, Q) = 1]|$, where $\mathbf{A}_r$ denotes the adversary $\mathbf{A}$ running with random tape r, a, b, c are chosen uniformly at random from $\mathbb{Z}_Q$ and all the multiplications are over $\mathbb{Z}_P^*$. The Integer Decisional Diffie-Hellman assumption (DDH) states that for every PPT $\mathbf{A}$, for every sequence $\{(P_k, Q_k, g_k)\}_k$ satisfying $|P_k| = k$ and $|Q_k| = \Theta(k)$, $\mathbf{Adv}_{\mathbf{A}}^{\mathsf{ddh}}(g_k, P_k, Q_k)$ is negligible in k.

Setup.

Let $p = rq + 1$ where $2^k < p < 2^{k+1}$, q is a large prime, and $\gcd(r, q) = 1$; let g generate $\mathbb{Z}_p^*$ and thus $\gamma = g^r$ generates the unique multiplicative subgroup of order q; let $\hat{r}$ be the least integer r such that $r\hat{r} = 1 \bmod q$. Assume $|m_0| = |m_1| < k/2$. Let $H : \{0,1\}^{2k} \times \mathbb{Z}_p \to \{0,1\}^{k/2}$ be a pairwise-independent family of hash functions. Define the randomized mapping $\phi : \langle\gamma\rangle \to \mathbb{Z}_p^*$ by $\phi(h) = h^{\hat{r}} g^{\beta q}$, for a uniformly chosen $\beta \in \mathbb{Z}_r$; notice that $\phi(h)^r = h$ and that for a uniformly chosen $h \in \langle\gamma\rangle$, $\phi(h)$ is a uniformly chosen element of $\mathbb{Z}_p^*$. The following protocol is a simple modification of the Naor-Pinkas 2-round oblivious transfer protocol [45]:

Construction 7.6. COT:

1. On input $\sigma \in \{0,1\}$, C chooses uniform $a, b \in \mathbb{Z}_q$, sets $c_\sigma = ab \bmod q$ and uniformly chooses $c_{1-\sigma} \in \mathbb{Z}_q$. C sets $x = \gamma^a$, $y = \gamma^b$, $z_0 = \gamma^{c_0}$, $z_1 = \gamma^{c_1}$ and sets $x' = \phi(x), y' = \phi(y), z'_0 = \phi(z_0), z'_1 = \phi(z_1)$. If the most significant bits of all of x', y', z'_0, z'_1 are 0, C sends the least significant k bits of each to S; otherwise C picks new $a, b, c_{1-\sigma}$ and starts over.

2. The sender recovers x, y, z_0, z_1 by raising to the power r, picks $f_0, f_1 \in H$ and then:

   - S repeatedly chooses uniform $r_0, s_0 \in \mathbb{Z}_q$ and sets $w_0 = x^{s_0}\gamma^{r_0}$, $w'_0 = \phi(w_0)$ until he finds a pair with $w'_0 \leq 2^k$. He then sets $K_0 = z_0^{s_0} y^{r_0}$.
   - S repeatedly chooses uniform $r_1, s_1 \in \mathbb{Z}_q$ and sets $w_1 = x^{s_1}\gamma^{r_1}$, $w'_1 = \phi(w_1)$ until he finds a pair with $w'_1 \leq 2^k$. He then sets $K_1 = z_1^{s_1} y^{r_1}$.

   S sends $w'_0 \| f_0 \| f_0(K_0) \oplus m_0 \| w'_1 \| f_1 \| f_1(K_1) \oplus m_1$

3. C recovers $K_\sigma = (w'_\sigma)^{rb}$ and computes m_σ.

Lemma 7.7. *S cannot distinguish between the case that C is following the COT protocol and the case that C is drawing from U_k; that is,*

$$V^S_{\mathrm{COT}}(m_0, m_1, \sigma) \approx V^S_{\mathrm{COT}:\mathcal{U}_C}(m_0, m_1, \sigma).$$

Proof. Suppose that there exists a distinguisher D with advantage ϵ. Then there exists a DDH adversary A with advantage at least $\epsilon/8 - \nu(k)$ for a negligible ν. A takes as input a triple $(\gamma^a, \gamma^b, \gamma^c)$, picks a random bit σ, sets $z_\sigma = \gamma^c$ and picks a uniform $z'_{1-\sigma} \in \{0,1\}^k$, and computes $x' = \phi(\gamma^a), y' = \phi(\gamma^b), z'_\sigma = \phi(z_\sigma)$; if all three are at most 2^k, then A outputs $D(x', y', z'_0, z'_1)$, otherwise A outputs 0.

Clearly, when $c \neq ab$,

$$\Pr[A(\gamma^a, \gamma^b, \gamma^c) = 1] \geq \frac{1}{8} \Pr[D(V^S_{\mathrm{COT}:\mathcal{U}_C}) = 1] \ ,$$

since the elements passed by A to D are uniformly chosen and D calls A with probability at least 1/8 (since each of x', y', z'_σ are greater than 2^k with probability at most 1/2). But when $c = ab$, then

$$\Pr[A(\gamma^a, \gamma^b, \gamma^c) = 1] \geq (1/8 - \nu(k)) \Pr[D(V^S_{\mathrm{COT}}) = 1] \ ,$$

since the elements passed by A to D are chosen exactly according to the distribution on C's output specified by COT; and since the probability that D is invoked by A is at least $1/8$ when $c \neq ab$ it can be at most $\nu(k)$ less when $c = ab$, by the Integer DDH assumption. Thus the DDH advantage of A is at least $\epsilon/8 - \nu(k)$. Since $\epsilon/8$ must be negligible by the DDH assumption, we have that D's advantage must also be negligible. $\square$

Lemma 7.8. When $m_0, m_1 \leftarrow U_{k/2}$, C cannot distinguish between the case that S is following the COT protocol and the case that S is sending uniformly chosen strings. That is, $V^C_{\mathrm{COT}}(U_{k/2}, U_{k/2}, \sigma) \approx V^C_{\mathrm{COT}:\mathcal{U}_S}(U_{k/2}, U_{k/2}, \sigma)$.

Proof. The group elements w_0, w_1 are uniformly chosen by S; thus when m_0, m_1 are uniformly chosen, the message sent by S must also be uniformly distributed. $\square$

Lemma 7.9. The COT protocol securely realizes the OT^2_1 functionality.

Proof. The protocol described by Pinkas and Naor is identical to the COT protocol, with the exception that ϕ is not applied to the group elements x, y, z_0, z_1, w_0, w_1 and these elements are not rejected if they are greater than 2^k. Suppose an adversarial sender can predict σ with advantage ϵ in COT; then he can be used to predict σ with advantage $\epsilon/16 - \nu(k)$ in the Naor-Pinkas protocol, by applying the map ϕ to the elements x, y, z_0, z_1 and predicting a coin flip if not all are less than 2^k, and otherwise using the sender's prediction against the message that COT would send. Likewise, any bit a chooser can predict about (m_0, m_1) with advantage ϵ in COT, can be predicted with advantage $\epsilon/4$ in the Naor-Pinkas protocol: the Chooser's message can be transformed into elements of $\langle\gamma\rangle$ by taking the components to the power r, and the resulting message of the Naor-Pinkas sender can be transformed by sampling from $w'_0 = \phi(w_0), w'_1 = \phi(w_1)$ and predicting a coin flip if either is greater than 2^k, but otherwise giving the prediction of the COT chooser on $w'_0 \| f_0 \| f_0(K_0) \oplus m_0 \| w'_1 \| f_1 \| f_1(K_1) \oplus m_1$. $\square$

Conjoining these three lemmas gives the following theorem:

Theorem 7.10. *Protocol COT covertly realizes the uniform-OT^2_1 functionality*

7.2.5 Combining The Pieces

We can combine the components developed up to this point to make a protocol which covertly realizes any two-party functionality. The final protocol, which we call COVERT-YAO, is simple: assume that both parties know a circuit C_f computing the functionality f. Bob first uses Yao's protocol to create a garbled circuit for $f(\cdot, x_B)$. Alice and Bob perform $|x_A|$ covert oblivious transfers for the garbled wire values corresponding to Alice's inputs. Bob sends the garbled gates to Alice. Finally, Alice collects the garbled output values and sends them to Bob, who de-garbles these values to obtain the output.

Theorem 7.11. *The* COVERT-YAO *protocol covertly realizes the functionality f.*

Proof. That (Alice, Bob) securely realize the functionality f follows from the security of Yao's protocol. Now consider the distribution of each message sent from Alice to Bob:

- In each execution of COT: each message sent by Alice is uniformly distributed
- Final values: these are masked by the uniformly chosen bits that Bob chose in garbling the output gates. To an observer, they are uniformly distributed.

Thus Bob's view, until the last round, is in fact identically distributed when Alice is running the protocol and when she is drawing from $\mathcal{U}$. Likewise, consider the messages sent by Bob:

- In each execution of COT: because the W_i^b from Yao's protocol are uniformly distributed, Theorem 7.10 implies that Bob's messages are indistinguishable from uniform strings.
- When sending the garbled circuit, the pseudorandomness of F and the uniform choice of the W_i^b imply that each garbled gate, even given one garbled input pair, is indistinguishable from a random string.

Thus Alice's view after all rounds of the protocol is indistinguishable from her view when Bob draws from $\mathcal{U}$.

If Bob can distinguish between Alice running the protocol and drawing from $\mathcal{B}$ after the final round, then he can also be used to distinguish between $f(X_A, x_B)$ and U_l. The approach is straightforward: given a candidate y, use the simulator from Yao's protocol to generate a view of the "data layer." If $y \leftarrow f(X_A, x_B)$, then, by the security of Yao's protocol, this view is indistinguishable from Bob's view when Alice is running the covert protocol. If $y \leftarrow U_l$, then the simulated view of the final step is distributed identically to Alice drawing from $\mathcal{U}$. Thus Bob's advantage will be preserved, up to a negligible additive term. $\square$

Notice that as the protocol COVERT-YAO is described, it is not secure against a malicious Bob who gives Alice a garbled circuit with different operations in the gates, which could actually output some constant message giving away Alice's participation even when the value $f(x_0, x_1)$ would not. If instead Bob sends Alice the masking values for the garbled output bits, Bob could still prevent Alice from learning $f(x_0, x_1)$ but could not detect her participation in the protocol in this way. We use this version of the protocol in the next section.

7.3 Fair Covert Two-party Computation Against Malicious Adversaries

The protocol presented in the previous section has two serious weaknesses. First, because Yao's construction conceals the function of the circuit, a malicious Bob can garble a circuit that computes some function other than the result Alice agreed to compute. In particular, the new circuit could give away Alice's input or output some distinguished string that allows Bob to determine that Alice is running the protocol. Additionally, the protocol is *unfair:* either Alice or Bob does not get the result.

In this section we present a protocol that avoids these problems. In particular, our solution has the following properties: (1) If both parties follow the protocol, both get the result; (2) If Bob cheats by garbling an incorrect circuit, neither party can tell whether the other is running the protocol, except with negligible advantage; and (3) Except with negligible probability, if one party terminates early and computes the result in time T, the other party can compute the result in time at most $O(T)$. Our protocol is secure in the random oracle model, under the Decisional Diffie Hellman assumption. We show at the end of this section, however, that our protocol can be made to satisfy a slightly weaker security condition without the use of a random oracle. (We note that the technique used in this section has some similarities to one that appears in [1].)

7.3.1 Definitions

We assume the existence of a non-interactive bitwise commitment scheme with commitments which are indistinguishable from random bits. One example is the (well-known) scheme which commits to b by $CMT(b; (r, x)) = r\|\pi(x)\|(x \cdot r) \oplus b$, where π is a one-way permutation on domain $\{0,1\}^k$, $x \cdot y$ denotes the inner-product of x and y over $GF(2)$, and $x, r \leftarrow U_k$. The integer DDH assumption implies the existence of such permutations.

Let f denote the functionality we wish to compute. We say that f is *fair* if for every distinguisher D_σ distinguishing $f(X_0, X_1)$ from U given X_σ with advantage at least ϵ, there is a distinguisher $D_{1-\sigma}$ with advantage at least $\epsilon - \nu(k)$, for a negligible function ν. (That is, if P_0 can distinguish $f(X_0, X_1)$ from uniform, so can P_1.) We say f is *strongly fair* if $(f(X_0, X_1), X_0) \approx (f(X_0, X_1), X_1)$.

A n-round, two-party protocol $\Pi = (P_0, P_1)$ to compute functionality f is said to be a strongly fair covert protocol for the bidirectional channel $\mathcal{B}$ if the following conditions hold:

- (External covertness): For any input $\bar{x}$, $\langle P_0(x_0), P_1(x_1)\rangle \approx \mathcal{B}$.
- (Strong Internal Covertness): There exists a PPT E (an *extractor*) such that if PPT $D(V)$ distinguishes between $V_{\Pi,i}^{P_\sigma}(\bar{x})$ and $V_{\Pi:\mathcal{B}_{1-\sigma},i}^{P_\sigma}(\bar{x})$ with advantage ϵ, $E^D(V_\Pi^{P_\sigma}(\bar{x}))$ computes $f(\bar{x})$ with probability at least $\epsilon/poly(k)$
- (Strong Fairness): If the functionality f is fair, then for any C_σ running in time T such that $\Pr[C_\sigma(V_{\Pi,i}^{\sigma}(\bar{x})) = f(\bar{x})] \geq \epsilon$, there exists a $C_{1-\sigma}$ running in time $O(T)$ such that $\Pr[C_{1-\sigma}(V_{\Pi,i}^{1-\sigma}(\bar{x})) = f(\bar{x})] = \Omega(\epsilon)$.

- (Final Covertness): For every PPT D there exists a PPT D' and a negligible ν such that for any x_σ and distribution $X_{1-\sigma}$, $\mathbf{Adv}_D^{V_\Pi^{P_\sigma}(X_{1-\sigma},x_\sigma),V_{\Pi:\mathcal{B}_{1-\sigma}}^{P_\sigma}(X_{1-\sigma},x_\sigma)}(k) \leq \mathbf{Adv}_{D'}^{f(X_{1-\sigma},x_\sigma),U_l}(k) + \nu(k)$.

Intuitively, the Internal Covertness requirement states that "Alice can't tell if Bob is running the protocol until she gets the answer," while Strong Fairness requires that "Alice can't get the answer unless Bob can." Combined, these requirements imply that neither party has an advantage over the other in predicting whether the other is running the protocol.

7.3.2 Construction

As before, we have two parties, P_0 (Alice) and P_1 (Bob), with inputs x_0 and x_1, respectively, and the function Alice and Bob wish to compute is $f : \{0,1\}^{l_0} \times \{0,1\}^{l_1} \rightarrow \{0,1\}^l$, presented by the circuit C_f. The protocol proceeds in three stages: COMMIT, COMPUTE, and REVEAL. In the COMMIT stage, Alice picks $k+2$ strings, r_0, and $s_0[0], \ldots, s_0[k]$, each k bits in length. Alice computes commitments to these values, using a bitwise commitment scheme which is indistinguishable from random bits, and sends the commitments to Bob. Bob does likewise (picking strings $r_1, s_1[0], \ldots, s_1[k]$).

The next two stages involve the use of a pseudorandom generator $G : \{0,1\}^k \rightarrow \{0,1\}^l$ which we will model as a random oracle *for the security argument only*: G itself must have an efficiently computable circuit. In the COMPUTE stage, Alice and Bob compute two serial runs ("rounds") of the covert Yao protocol described in the previous section. If neither party cheats, then at the conclusion of the COMPUTE stage, Alice knows $f(x_0, x_1) \oplus G(r_1)$ and Bob's value $s_1[0]$; while Bob knows $f(x_0, x_1) \oplus G(r_0)$ and Alice's value $s_0[0]$. The REVEAL stage consists of k rounds of two runs each of the covert Yao protocol. At the end of each round i, if nobody cheats, Alice learns the i^{th} bit of Bob's string r_1, labeled $r_1[i]$, and also Bob's value $s_1[i]$, and Bob learns $r_0[i]$, $s_0[i]$. After k rounds in which neither party cheats, Alice thus knows r_1 and can compute $f(x_0, x_1)$ by computing the exclusive-or of $G(r_1)$ with the value she learned in the COMPUTE stage, and Bob can likewise compute the result.

Each circuit sent by Alice must check that Bob has obeyed the protocol; thus at every round of every stage, the circuit that Alice sends to Bob takes as input the opening of all of Bob's commitments, and checks to see that all of the bits Alice has learned so far are consistent with Bob's input. The difficulty to overcome with this approach is that the result of the check cannot be returned to Alice without giving away that Bob is running the protocol. To solve this problem, Alice's circuits also take as input the last value $s_0[i-1]$ that Bob learned. If Alice's circuit ever finds that the bits she has learned are inconsistent with Bob's input, or that Bob's input for $s_0[i-1]$ is not consistent with the actual value of $s_0[i-1]$, the output is a uniformly chosen string of the appropriate length. Once this happens, all future outputs to Bob will also be independently and uniformly chosen, because he will have the wrong value for $s_0[i]$, which will give him the wrong value for $s_0[i+1]$, etc. Thus the values $s_0[1, \ldots, k]$ serve as "state" bits that Bob maintains for Alice. The analogous statements hold for Bob's circuits and Alice's inputs.

Construction 7.12. (Fair covert two-party computation)

Inputs and setup. To begin, each party P_σ chooses $k+2$ random strings r_σ, $s_\sigma[0]$,...,$s_\sigma[k] \leftarrow U_k$. P_σ's inputs to the protocol are then $X_\sigma = (x_\sigma, r_\sigma, s_\sigma[0 \ldots k])$.

COMMIT stage. Each party P_σ computes the commitment $\kappa_\sigma = CMT(X_\sigma; \rho_\sigma)$ and sends this commitment to the other party. Denote by K_σ the value that P_σ interprets as a commitment to $X_{1-\sigma}$, that is, K_0 denotes the value Alice interprets as a commitment to Bob's input X_1.

COMPUTE stage. The COMPUTE stage consists of two serial runs of the COVERT-YAO protocol.

1. Bob garbles the circuit COMPUTE$_1$ shown in figure 7.1, which takes x_0, r_0, $s_0[0], \ldots, s_0[k]$, and ρ_0 as input and outputs $G(r_1) \oplus f(x_0, x_1) \| s_1[0]$ if K_1 is a commitment to X_0. If this check fails, COMPUTE$_1$ outputs a uniformly chosen string, which has no information about $f(x_0, x_1)$ or $s_1[0]$. Bob and Alice perform the COVERT-YAO protocol; Alice labels her result $F_0 \| S_0[0]$.

2. Alice garbles the circuit COMPUTE$_0$ shown in figure 7.1, which takes x_1, r_1,

$\text{COMPUTE}_\sigma(x_{1-\sigma}, r, s[0 \ldots k], \rho) =$	$\text{REVEAL}^i_\sigma(x_{1-\sigma}, S_{1-\sigma}[i-1], r, s_{1-\sigma}[0 \ldots k], \rho) =$
if $(K_\sigma = CMT(x_{1-\sigma}, r, s; \rho))$	Let $F = G(r) \oplus f(x_0, x_1)$
then set $F = G(r_\sigma) \oplus f(x_0, x_1)$	**if** $(K_\sigma = CMT(x_{1-\sigma}, r, s_{1-\sigma}; \rho)$ **and**
set $S = s_\sigma[0]$ **else** draw $F \leftarrow U_l$,	$F = F_\sigma$ **and**
draw $S \leftarrow U_k$.	$R_\sigma[i-1] = r[i-1]$ **and**
output $F \| S$	$S_{1-\sigma}[i-1] = s_\sigma[i-1]$ **and**
	$S_\sigma[i-1] = s_{1-\sigma}[i-1])$ **then**
	set $R = r_\sigma[i], S = s_\sigma[i]$
	else draw $R \leftarrow \{0,1\}$, $S \leftarrow U_k$
	output $R \| S$

Figure 7.1: The circuits COMPUTE and REVEAL.

$s_1[0], \ldots, s_1[k]$, and ρ_1 as input and outputs $G(r_0) \oplus f(x_0, x_1) \| s_0[0]$ if K_0 is a commitment to X_1. If this check fails, COMPUTE_0 outputs a uniformly chosen string, which has no information about $f(x_0, x_1)$ or $s_0[0]$. Bob and Alice perform the COVERT-YAO protocol; Bob labels his result $F_1 \| S_1[0]$.

REVEAL stage. The REVEAL stage consists of k rounds, each of which consists of 2 runs of the COVERT-YAO protocol:

1. in round i, Bob garbles the circuit REVEAL^i_1 shown in figure 7.1, which takes input $x_0, S_0[i-1], r_0, s_0[0 \ldots k], \rho_0$ and checks that:
   - Bob's result from the COMPUTE stage, F_1, is consistent with x_0, r_0.
   - The bit $R_1[i-1]$ which Bob learned in round $i-1$ is equal to bit $i-1$ of Alice's secret r_0. (By convention, and for notational uniformity, we will define $R_0[0] = R_1[0] = r_0[0] = r_1[0] = 0$)
   - The state $S_0[i-1]$ that Bob's circuit gave Alice in the previous round was correct. (Meaning Alice obeyed the protocol up to round $i-1$)
   - Finally, that the state $S_1[i-1]$ revealed to Bob in the previous round was the state $s_0[i-1]$ which Alice committed to in the COMMIT stage.

If all of these checks succeed, Bob's circuit outputs bit i of r_1 and state $s_1[i]$; otherwise the circuit outputs a uniformly chosen $k+1$-bit string. Alice and Bob perform COVERT-YAO and Alice labels the result $R_0[i], S_0[i]$.

2. Alice garbles the circuit REVEAL_0^i depicted in figure 7.1 which performs the analogous computations to REVEAL_1^i, and performs the COVERT-YAO protocol with Bob. Bob labels the result $R_1[i], S_1[i]$.

After k such rounds, if Alice and Bob have been following the protocol, we have $R_1 = r_0$ and $R_0 = r_1$ and both parties can compute the result. The "states" s are what allow Alice and Bob to check that all previous outputs and key bits (bits of r_0 and r_1) sent by the other party have been correct, without ever receiving the results of the checks or revealing that the checks fail or succeed.

Theorem 7.13. *Construction 7.12 is a strongly fair covert protocol realizing the functionality f*

Proof. The correctness of the protocol follows by inspection. The two-party security follows by the security of Yao's protocol. Now suppose that some party, wlog Alice, cheats (by sending a circuit which computes an incorrect result) in round j. Then, the key bit $R_0[j+1]$ and state $S_0[j+1]$ Alice computes in round $j+1$ will be randomized; and with overwhelming probability every subsequent result that Alice computes will be useless. Assuming Alice can distinguish $f(x_0, X_1)$ from uniform, she can still compute the result in at most 2^{k-j} time by exhaustive search over the remaining key bits. By successively guessing the round at which Alice began to cheat, Bob can compute the result in time at most 2^{k-j+2}. If Alice aborts at round j, Bob again can compute the result in time at most 2^{k-j+1}. If Bob cheats in round j by giving inconsistent inputs, with high probability all of his remaining outputs are randomized; thus cheating in this way gives him no advantage over aborting in round $j-1$. Thus, the fairness property is satisfied.

If G is a random oracle, neither Alice nor Bob can distinguish anything in their view from uniformly chosen bits without querying G at the random string chosen by the other. So given a distinguisher D running in time $p(k)$ for $V_{\Pi,i}^{P_0}(\bar{x})$ with advantage ϵ, it is simple to write an extractor which runs D, recording its queries to G, picks one such query (say, q) uniformly, and outputs $G(q) \oplus F_0$. Since D can only have an advantage when it queries r_1, E will pick $q = r_1$ with probability at least $1/p(k)$ and in this case correctly outputs $f(x_0, x_1)$. Thus the Strong Internal Covertness property is satisfied. $\square$

Weakly fair covertness.

We can achieve a slightly weaker version of covertness without using random oracles. Π is said to be a *weakly fair* covert protocol for the channel $\mathcal{B}$ if Π is externally covert, and has the property that if f is strongly fair, then for every distinguisher D_σ for $V_{\Pi,i}^{P_\sigma}(\bar{x})$ with significant advantage ϵ, there is a distinguisher $D_{1-\sigma}$ for $V_{\Pi,i}^{P_{1-\sigma}}(\bar{x})$ with advantage $\Omega(\epsilon)$. Thus in a weakly fair covert protocol, we do not guarantee that both parties get the result, only that if at some point in the protocol, one party can tell that the other is running the protocol with significant advantage, the same is true for the other party.

We note that in the above protocols, if the function G is assumed to be a pseudorandom generator (rather than a random oracle), then the resulting protocol exhibits weakly fair covertness. Suppose D_σ has significant advantage ϵ after round j, as in the hypothesis of weak covertness. Notice that given $r_{1-\sigma}[1 \ldots j-1], G(r_{1-\sigma}) \oplus f(\bar{x})$, the remainder of P_σ's view can be simulated efficiently. Then D_σ must be a distinguisher for $G(r)$ given the first $j-1$ bits of r. But since f is strongly fair, $P_{1-\sigma}$ can apply D_σ to $G(r_\sigma) \oplus f(\bar{x})$ by guessing at most 1 bit of r_σ and simulating P_σ's view with his own inputs. Thus $P_{1-\sigma}$ has advantage at least $\epsilon/2 - \nu(k) = \Omega(\epsilon)$.

Chapter 8

Future Research Directions

While this thesis has resolved several of the open questions pertaining to universal steganography, there are still many interesting open questions about theoretical steganography. In this section we highlight those that seem most important.

8.1 High-rate steganography

We have shown that for a universal blockwise stegosystem with bounded sample access to a channel, the optimal rate is bounded above by both the minimum entropy of the channel and the logarithm of the sample bound. Three general research directions arise from this result. First, a natural question is what happens to this bound if we remove the universality and blockwise constraints. A second natural direction to pursue is the question of efficiently detecting the use of a stegosystem that exceeds the maximum secure rate. A third interesting question to explore is the relationship between extractors and stegosystems.

If we do not restrict ourselves to consider universal blockwise stegosystems, there is some evidence to suggest that it is possible to achieve a much higher rate. For instance, for the uniform channel $\mathcal{U}$, the $\mathsf{IND\$\text{-}CPA}$ encryption scheme in section 2 has rate which converges to 1. Likewise, a recent proposal by Van Le [41] describes a stegosystem based on the "folklore" observation that perfect compression for a channel yields secure steganography; the system described there is not universal, nor is it secure in a blockwise model, but the rate approaches the Shannon entropy for any efficiently sampleable channel with entropy bounded by the logarithm of the security parameter k. Thus it is natural to wonder whether there is a reasonable security model and a reasonable class of nonuniversally accessible stegosystems which are provably secure under this model, yet have rate which substantially exceeds that of the construction in Chapter 6.

We show that any blockwise stegosystem which exceeds the minimum entropy can be detected by giving a detection algorithm which draws many samples from the channel. It is an interesting question whether the number of samples required can be reduced significantly for some channels. It is not hard to see that artificial channels can be designed for which this is the case using, for instance, a trapdoor permutation for which the warden knows the trapdoor. However, a more natural example would be of interest.

The design methodology of (blockwise) transforming the uniform channel to an arbitrary channel, as well as the minimum entropy upper bound on the rate of a stegosystem suggest that there is a connection to *extractors*. An extractor is a function that transforms a sample from an arbitrary blockwise source of minimum entropy m and a short random string into a string of roughly m bits that has distribution statistically close to uniform. (In fact a universal hash function is an extractor.) It would be interesting to learn whether there is any deeper connection between stegosystems and extractors, for instance, the *decoding* algorithm for a stegosystem (SE, SD) acts as an extractor-like function for some distributions; in particular $SD_K(\cdot)$ optimally extracts entropy from the distribution $SE_K(U)$. However, it is not immediately obvious how to extend this to a general extractor.

8.2 Public Key Steganography

The necessary and sufficient conditions for the existence of a public-key stegosystem constitute an open question. Certainly for a *universal* stegosystem the necessary and sufficient condition is the existence of a trapdoor predicate family with domains that are computationally indistinguishable from a polynomially dense set: as we showed in

Chapter 4, such primitives are sufficient for IND\$-CPA public-key encryption; while on the other hand, the existence of a universal public-key stegosystem implies the existence of a public-key stegosystem for the uniform channel, which is by itself a trapdoor predicate family with domains that are computationally indistinguishable from a set of density 1. Unlike the case with symmetric steganography, however, we are not aware of a reduction from a stegosystem for an arbitrary channel to a dense-domain trapdoor predicate family.

In a similar manner, it is an open question whether steganographic key exchange protocols can be constructed based on intractability assumptions other than the Decisional Diffie-Hellman assumption. This is in contrast to cryptographic key exchange, which is implied by the existence of any public-key encryption scheme or oblivious transfer protocol. It is not clear whether the existence of IND\$-CPA public-key encryption implies the existence of SKE protocols.

8.3 Active attacks

Concerning steganography in the presence of active attacks, several questions remain open. Some standard cryptographic questions remain about chosen-covertext security, and substitution-robust steganography. A more important issue is a model of a disrupting adversary which more closely models the type of attacks applied to existing proposals in the literature for robust stegosystems.

There are several open cryptographic questions relating to chosen-covertext security. For example, it is not clear whether IND\$-CCA-secure public-key encryption schemes exist in the standard model (without random oracles). As we alluded to in chapter 5, all of the known general constructions of chosen-ciphertext secure encryption schemes are easily distinguished from random bits, and the known schemes depending on specific intractability assumptions seem to depend on using testable subgroups. Another interesting question is whether chosen covertext security can be achieved with oracle-only access to the channel. The key problem here is in ensuring that it is hard to find more than one valid encoding of a valid ciphertext; this seems difficult to accomplish without repeatable access to the channel. To avoid this problem, Backes and Cachin [7] have introduced the notion of Replayable Chosen Covertext (RCCA) security, which is identical to sCCA security, with the exception that the adversary is forbidden to submit covertexts which decode to the challenge hiddentext. The problem with this approach is that the replay attack seems to be a viable attack in the real world. Thus it is an interesting question to investigate the possibility of notions "in-between" sCCA and RCCA.

Similar questions about substitution-robustness remain open. It is an interesting problem to design a universal provably-secure substitution robust stegosystem that requires only sender access to the channel. Also of interest is whether the requirement that Bob can evaluate an admissible superset of the relation R can be removed. Intuitively, it seems that the ability to evaluate R is necessary for substitution robustness, because the decoding algorithm evaluates R to an extent: if $R(x, y)$, then it should be the case that $SD(x) = SD(y)$ except with negligible probability. The trouble with this intuition is first, that there is no requirement that decoding a single document should return anything meaningful, and second, that while such an algorithm evaluates a superset of R, it may not be admissible. In light of our proof that no stegosystem can be secure against both distinguishing and disrupting adversaries, it is also interesting to investigate the possibility of substitution robustness against adversaries with access to a decoding oracle.

The most important open question concerning robust steganography is the mismatch between substitution robustness and the types of attacks perpetrated against typical proposals for robust steganography. Such attacks include strategies such as splitting a single document into a series of smaller documents with the same meaning, merging two or more documents into a single document with the same meaning, and reordering documents in a list. Especially if there is no bound on the length of sequences to which these operations can be applied, it seems difficult to even write a general description of the rules such a warden must follow; and although it is reasonably straightforward to counteract any single attack in the previous list, composing several of them with relation-bounded substitutions as well seems to lead to attacks which are difficult to defend against.

8.4 Covert Computation

In the area of covert computation, this thesis leaves room for improvement and open problems. For example, can (strongly) fair covert two-party computation secure against malicious adversaries be satisfied without random oracles? It seems at least plausible that constructions based on concrete assumptions such as the "knowledge-of-exponent" assumption or the "generalized BBS" assumption may allow construction of such protocols, yet the obvious applications always destroy the final covertness property. A related question is whether covert two-party computation can be based on general cryptographic assumptions rather than the specific Decisional Diffie-Hellman assumption used here.

Another open question is that of improving the efficiency of the protocols presented here, either by designing protocols for specific goals or through adapting efficient two-party protocols to provide covertness. A possible direction to pursue would be "optimistic" fairness involving a trusted third party. In this case, though, there is the question of how the third party could "complete" the computation without revealing participation.

Another interesting question is whether the notion of covert two-party computation can be extended in some natural and implementable way to multiple parties. Such a generalization could have important applications in the area of anonymous communications, allowing, for instance, the deployment of undetectable anonymous remailer networks. The difficulty here is in finding a sensible model - how can a multiparty computation take place without knowing who the other parties are? If the other parties are to be known, how can their participation be secret? What if the normal communication patterns between parties is not the complete graph? In addition to these difficulties, the issues associated with cheating players become more complex, and there seems to be no good candidate protocol for the uniform channel.

8.5 Other models

The results of Chapter 3 show that the ability to sample from a channel in our model is necessary for steganographic communication using that channel. Since in many cases we do not understand the channel well enough to sample from it, a natural question is whether there exist models where less knowledge of the distribution is necessary; such a model will necessarily restrict the adversary's knowledge of the channel as well. One intuition is that typical steganographic adversaries are not monitoring the traffic between a specific pair of individuals in an effort to confirm suspicious behavior, but are monitoring a high-volume stream of traffic between many points looking for the "most suspicious" behavior; so stegosystems which could be detected by analyzing a long sequence of communications might go undetected if only single messages are analyzed. This type of model is tantalizing because there are unconditionally secure cryptosystems under various assumptions about adversaries with bounded storage [18, 50], but it remains an interesting challenge to give a satisfying formal model and provably secure construction for this scenario.

Bibliography

[1] G. Aggarwal, N. Mishra and B. Pinkas. Secure computation of the k'th-ranked element To appear in *Advances in Cryptology – Proceedings of Eurocrypt '04*, 2004.

[2] Luis von Ahn, Manuel Blum and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI.

[3] Luis von Ahn and Nicholas J. Hopper. Public-Key Steganography. Submitted to CRYPTO 2003.

[4] L. von Ahn and N. Hopper. Public-Key Steganography. To appear in *Advances in Cryptology – Proceedings of Eurocrypt '04*, 2004.

[5] Ross J. Anderson and Fabien A. P. Petitcolas. *On The Limits of Steganography.* IEEE Journal of Selected Areas in Communications, 16(4). May 1998.

[6] Ross J. Anderson and Fabien A. P. Petitcolas. *Stretching the Limits of Steganography.* In: *Proceedings of the first International Information Hiding Workshop.* 1996.

[7] M. Backes and C. Cachin. Public-Key Steganography with Active Attacks. *IACR e-print archive report 2003/231*, 2003.

[8] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In: *Advances in Cryptology – Proceedings of CRYPTO 98*, pages 26–45, 1998.

[9] M. Bellare and P. Rogaway. Random Oracles are Practical. *Computer and Communications Security: Proceedings of ACM CCS 93*, pages 62–73, 1993.

[10] M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. *Advances in Cryptology – Proceedings of CRYPTO '89*, pages 547-557, 1990.

[11] E.R Berlekamp. Bounded Distance +1 Soft-Decision Reed-Solomon Decoding. *IEEE Transactions on Information Theory*, 42(3), pages 704–720, 1996.

[12] J. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman. Hiding Information in Documents Images. In: Conference on Information Sciences and Systems, 1995.

[13] M. Blum and S. Goldwasser. An Efficient Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information. *Advances in Cryptology: CRYPTO 84*, Springer LNCS 196, pages 289-302. 1985.

[14] M. Blum and S. Micali. How to generate cryptographically strong sequences of random bits. In: *Proceedings of the 21st FOCS*, pages 112–117, 1982.

[15] E. Brickell, D. Chaum, I. Damgärd, J. van de Graaf: Gradual and Verifiable Release of a Secret. *Advances in Cryptology – Proceedings of CRYPTO '87*, pages 156-166, 1987.

[16] C. Cachin. *An Information-Theoretic Model for Steganography.* In: *Information Hiding – Second International Workshop, Preproceedings.* April 1998.

[17] C. Cachin. *An Information-Theoretic Model for Steganography.* In: *Information and Computation* 192 (1): pages 41–56, July 2004.

[18] C. Cachin and U. Maurer. Unconditional Security Against Memory-Bounded Adversaries. In: *Advances in Cryptology – CRYPTO '97*, Springer LNCS 1294, pp. 292–306, 1997.

[19] R. Canetti, U. Feige, O. Goldreich and M. Naor. Adaptively Secure Multi-party Computation. *28th Symposium on Theory of Computing (STOC 96)*, pages 639-648. 1996.

[20] R. Cramer and V. Shoup. A practical public-key cryptosystem provably secure against adaptive chosen ciphertext attack. *Advances in Cryptology: CRYPTO 98*, Springer LNCS 1462, pages 13-27, 1998.

[21] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. *Advances in Cryptology: EUROCRYPT 2002*, Springer LNCS 2332, pages 45-64. 2002.

[22] S. Craver. *On Public-Key Steganography in the Presence of an Active Warden.* In: *Information Hiding – Second International Workshop, Preproceedings.* April 1998.

[23] D. Dolev, C. Dwork, and M. Naor. Non-malleable Cryptography. *23rd Symposium on Theory of Computing (STOC '91)*, pages 542-552. 1991.

[24] Z. Galil, S. Haber, M. Yung. Cryptographic Computation: Secure Fault-Tolerant Protocols and the Public-Key Model. *Advances in Cryptology – Proceedings of CRYPTO '87*, pages 135-155, 1987.

[25] O. Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, 2001.

[26] O. Goldreich. Secure Multi-Party Computation. Unpublished Manuscript. *http://philby.ucsd.edu/books.html*, 1998.

[27] O. Goldreich, S. Goldwasser and S. Micali. How to construct pseudorandom functions. *Journal of the ACM*, vol 33, 1998.

[28] O. Goldreich and L.A. Levin. A Hardcore predicate for all one-way functions. In: *Proceedings of 21st STOC*, pages 25–32, 1989.

[29] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game. *Nineteenth Annual ACM Symposium on Theory of Computing*, pages 218-229.

[30] S. Goldwasser and M. Bellare. Lecture Notes on Cryptography. Unpublished manuscript, August 2001. available electronically at `http://www-cse.ucsd.edu/~mihir/pa pers/gb.html`.

[31] S. Goldwasser and S. Micali. Probabilistic Encryption & how to play mental poker keeping secret all partial information. In: *Proceedings of the 14th STOC*, pages 365–377, 1982.

[32] D. Gruhl, W. Bender, and A. Lu. Echo Hiding. In: *Information Hiding: First International Workshop*, pages 295–315, 1996.

[33] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4), pages 1364-1396, 1999.

[34] N. Hopper, J. Langford and L. Von Ahn. Provably Secure Steganography. *Advances in Cryptology – Proceedings of CRYPTO '02*, pages 77-92, 2002.

[35] Nicholas J. Hopper, John Langford, and Luis von Ahn. *Provably Secure Steganography.* CMU Tech Report CMU-CS-TR-02-149, 2002.

[36] Russell Impagliazzo and Michael Luby. *One-way Functions are Essential for Complexity Based Cryptography.* In: 30th FOCS, November 1989.

[37] G. Jagpal. *Steganography in Digital Images* Thesis, Cambridge University Computer Laboratory, May 1995.

[38] D. Kahn. *The Code Breakers.* Macmillan 1967.

[39] J. Katz and M. Yung. Complete characterization of security notions for probabilistic private-key encryption. In: *Proceedings of 32nd STOC*, pages 245–254, 1999.

[40] Stefan Katzenbeisser and Fabien A. P. Petitcolas. *Information hiding techniques for steganography and digital watermarking.* Artech House Books, 1999.

[41] T. Van Le. Efficient Provably Secure Public Key Steganography *IACR e-print archive report 2003/156*, 2003.

[42] Y. Lindell. A Simpler Construction of CCA2-Secure Public Key Encryption. *Advances in Cryptology: EUROCRYPT 2003*, Springer LNCS 2656, pages 241-254. 2003.

[43] K. Matsui and K. Tanaka. *Video-steganography.* In: *IMA Intellectual Property Project Proceedings*, volume 1, pages 187–206, 1994.

[44] T. Mittelholzer. *An Information-Theoretic Approach to Steganography and Watermarking* In: *Information Hiding – Third International Workshop.* 2000.

[45] M. Naor and B. Pinkas. Efficient Oblivious Transfer Protocols. In: *Proceedings of the 12th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 448–457. 2001.

[46] M. Naor, B. Pinkas and R. Sumner. Privacy Preserving Auctions and Mechanism Design. In: *Proceedings, 1999 ACM Conference on Electronic Commerce.*

[47] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. *21st Symposium on Theory of Computing (STOC 89)*, pages 33-43. 1989.

[48] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. *22nd Symposium on Theory of Computing (STOC 90)*, pages 427-437. 1990.

[49] C. Neubauer, J. Herre, and K. Brandenburg. Continuous Steganographic Data Transmission Using Uncompressed Audio. In: *Information Hiding: Second International Workshop*, pages 208–217, 1998.

[50] N. Nissan. Pseudorandom generators for space-bounded computation. *Combinatorica* 12(1992):449–461.

[51] B. Pinkas. Fair Secure Two-Party Computation. In: *Advances in Cryptology – Eurocrypt '03*, pp 87–105, 2003.

[52] C. Rackoff and D. Simon. Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *Advances in Cryptology: CRYPTO 91*, Springer LNCS 576, pages 433-444, 1992.

[53] L. Reyzin and S. Russell. Simple Stateless Steganography. IACR e-print archive report 2003/093, 2003.

[54] Phillip Rogaway, Mihir Bellare, John Black and Ted Krovetz. *OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption*. In: *Proceedings of the Eight ACM Conference on Computer and Communications Security (CCS-8)*. November 2001.

[55] J. Rompel. One-way functions are necessary and sufficient for secure signatures. *22nd Symposium on Theory of Computing (STOC 90)*, pages 387-394. 1990.

[56] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. *40th IEEE Symposium on Foundations of Computer Science (FOCS 99)*, pages 543-553. 1999.

[57] J. A. O'Sullivan, P. Moulin, and J. M. Ettinger *Information theoretic analysis of Steganography*. In: *Proceedings ISIT '98*. 1998.

[58] C.E. Shannon. *Communication theory of secrecy systems*. In: *Bell System Technical Journal*, 28 (1949), pages 656-715.

[59] G.J. Simmons. *The Prisoner's Problem and the Subliminal Channel*. In: *Proceedings of CRYPTO '83*. 1984.

[60] L. Welch and E.R. Berlekamp. Error correction of algebraic block codes. US Patent Number 4,663,470, December 1986.

[61] A. Westfeld, G. Wolf. *Steganography in a Video Conferencing System*. In: *Information Hiding – Second International Workshop, Preproceedings.* April 1998.

[62] J. Wolfowitz. *Coding Theorems of Information Theory.* Springer-Verlag, Berlin and Prentice-Hall, NJ, 1978.

[63] A. C. Yao. Protocols for Secure Computation. *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, 1982, pages 160–164.

[64] A. C. Yao. How to Generate and Exchange Secrets. *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 1986, pages 162–167.

[65] A. Young and M. Yung. Kleptography: Using Cryptography against Cryptography. *Advances in Cryptology: Eurocrypt 87*, Springer LNCS 1233, pages 62-74, 1987.

[66] J Zollner, H.Federrath, H.Klimant, A.Pftizmann, R. Piotraschke, A.Westfield, G.Wicke, G.Wolf. *Modeling the security of steganographic systems*. In: *Information Hiding – Second International Workshop, Preproceedings.* April 1998.