Randomness-optimal Steganography

Aggelos Kiayias∗ Alexander Russell†
akiayias@cse.uconn.edu acr@cse.uconn.edu

Narasimha Shashidhar
karpoo@cse.uconn.edu

Department of Computer Science and Engineering
University of Connecticut, Storrs, CT

Abstract

Steganographic protocols enables one to “embed” covert messages into inconspicuous data over a public communication channel in such a way that no one, aside from the sender and the intended receiver can even detect the presence of the secret message. In this paper, we provide a new provably-secure, private-key steganographic encryption protocol. We prove the security of our protocol in the complexity-theoretic framework where security is quantified as the advantage (compared to a random guess) that the adversary has in distinguishing between innocent covertext and stegotext that embeds a message of his choice. The fundamental building block of our steganographic encryption protocol is a “one-time stegosystem” that allows two parties to transmit messages of length at most that of the shared key with information-theoretic security guarantees. The employment of a pseudorandom generator (PRG) permits secure transmission of longer messages in the same way that such a generator allows the use of one-time pad encryption for messages longer than the key in symmetric encryption. In this paper, we initiate the study of employing randomness extractors in a steganographic protocol construction to embed secret messages over the channel. To the best of our knowledge this is the first time randomness extractors have been applied in steganography.

Keywords: Information hiding, steganography, data hiding, steganalysis, covert communication.

1 Introduction

The steganographic communication problem can be best described using Simmons [13] formulation of the problem - In this scenario, prisoners Alice and Bob wish to communicate securely in the presence of an adversary, called the “Warden,” who monitors whether they exchange “conspicuous” messages. In particular, Alice and Bob may exchange messages that adhere to a certain channel distributions that represents “inconspicuous” communication. By controlling the messages that are transmitted over such a channel, Alice and Bob may exchange messages that cannot be detected by the Warden. There have been two approaches in formalizing this problem, one based on information theory [1, 15, 5] and one based on complexity theory [3, 4]. The latter approach is more concrete and has the potential of allowing more efficient constructions. Most steganographic constructions supported by provable security guarantees are instantiations of the following basic procedure (often referred to as “rejection-sampling”).

The problem specifies a family of message distributions (the “channel distributions”) that provide a number of possible options for a so-called “covertext” to be transmitted. Additionally, the sender and the
receiver possess some sort of private information (typically a keyed hash function, MAC, or other similar function) that maps channel messages to a single bit. In order to send a message bit \( m \), the sender draws a covertext from the channel distribution, applies the function to the covertext and checks whether it happens to produce the “stegotext” \( m \) he originally wished to transmit. If this is the case, the covertext is transmitted. In case of failure, this procedure is repeated. While this is a fairly concrete procedure, there are a number of choices to be made with both practical and theoretical significance. From the security viewpoint, one is primarily interested in the choice of the function that is shared between the sender and the receiver. From a practical viewpoint, one is primarily interested in how the channel is implemented and whether it conforms to the various constraints that are imposed on it by the steganographic protocol specifications (e.g., are independent draws from the channel allowed? does the channel remember previous draws? etc.).

Our model differs from the traditional approach to steganography where the sender modifies a covertext that is known to the adversary in an effort to embed secret data. Such an approach is secure only against adversaries with limited detection capability. This approach is found, for instance, in several software applications which manipulate certain pixels of visual images to embed hidden information. While such minor perturbations to an image may be imperceptible to the human eye, it is trivially discerned by an algorithm with access to the original cover image.

As mentioned above, the security of a stegosystem can be naturally phrased in information-theoretic terms (cf. [1]) or in complexity-theoretic terms [3]. Informally, the latter approach considers the following experiment for the warden-adversary: The adversary selects a message to be embedded and receives either covertexts that embed the message or covertexts simply drawn from the channel distribution (without any embedding). The adversary is then asked to distinguish between the two cases. Clearly, if the probability of success is very close to 1/2 it is natural to claim that the stegosystem provides security against such (eavesdropping) adversarial activity. FORMULATION OF STRONGER ATTACKS (SUCH AS ACTIVE ATTACKS) IS ALSO POSSIBLE.

Given the above framework, Hopper et al. [3] provided a provably secure stegosystem that pairs rejection sampling with a pseudorandom function family. In this article we take an alternative approach to the design of provably secure stegosystems. Our main contribution is the design of a building block that we call a one-time stegosystem: this is a steganographic protocol that is meant to be used for a single message transmission and is proven secure in an information-theoretic sense, provided that the key that is shared between the sender and the receiver is of sufficient length. In particular we show that we can securely transmit a \( \nu \) bit message with a secret key of length \( \nu \); Our basic building block is a natural analogue of a one time-pad for steganography. It is based on the rejection sampling technique outlined above in combination with randomness extractors. To the best of our knowledge, this is the first time randomness extractors have been employed in the design of steganographic protocols. Given a one-time stegosystem, it is fairly straightforward to construct provably secure steganographic encryption for longer messages by using a pseudorandom generator (PRG) to stretch a random seed that is shared by the sender and the receiver to sufficient length. The resulting stegosystem is provably secure in the computational model.

## 2 Definitions and Tools

### 2.1 Preliminaries

We say that a function \( \mu : \mathbb{N} \rightarrow \mathbb{R} \) is \textit{negligible} if for every positive polynomial \( p(\cdot), \) there exists an \( N \) such that for all \( n > N, \mu(n) < \frac{1}{p(n)} \). We use the notation \( x \leftarrow X \) to denote sampling an element \( x \) from a distribution \( X \) and the notation \( x \leftarrow R S \) to denote sampling an element \( x \) uniformly at random from a set \( S \). For a function \( f \) and a distribution \( X \) on its domain, \( f(X) \) denotes the distribution of sampling \( x \) from \( X \) and applying \( f \) to \( x \). The uniform distribution on \( \{0,1\}^d \) is denoted by \( U_d \) and \( U(X) \) denotes the uniform distribution on a finite set \( X \). We denote the length (in bits) of a string or integer \( s \) by \( |s| \) and the cardinality of a set \( S \) is denoted by \( |S| \). The concatenation of string \( s_1 \) and string \( s_2 \) is denoted by \( s_1 \circ s_2 \). “log” indicates the logarithm base 2 and “ln” denotes the natural logarithm. For completeness, we record below a few inequalities we use.
**Theorem 1** (Markov’s inequality). Let $X$ be a random variable that takes only non-negative real values. Then for every $\alpha > 0$, we have:

$$\Pr[X \geq \alpha] \leq \frac{\mathbb{E}[X]}{\alpha}.$$ 

**Theorem 2** (Law of Total Probability). Let $A$ and $B$ be events in a probability space $\Omega$, and $0 < \Pr[B] < 1$. Then,

$$\Pr[A] = \Pr[A|B] \cdot \Pr[B] + \Pr[A|\overline{B}] \cdot \Pr[\overline{B}].$$ 

**Theorem 3** (Boole’s inequality). Let $A_1, A_2, \ldots, A_m$ be a countable set of events in a probability space $\Omega$. Then,

$$\Pr[\bigcup_{i=1}^{m} A_i] \leq \sum_{i=1}^{m} \Pr[A_i].$$ 

### 2.2 $\epsilon$-biased functions

**Definition 1** ([14]). Let $P$ be a distribution with a finite support $X$. A function $f : X \to Y$ is $\epsilon$-biased if

$$|\Pr_{x \leftarrow P}[f(x) = y] - 1/|Y|| < \epsilon \ \forall \ y \in Y.$$ 

We say that $f$ is unbiased if $f$ is $\epsilon$-biased for $\epsilon$ a negligible function of the appropriate security parameter and finally $f$ is said to be perfectly unbiased if

$$|\Pr_{x \leftarrow P}[f(x) = y] - 1/|Y|| = 0 \ \forall \ y \in Y.$$ 

### 2.3 min-entropy

We use min-entropy to quantify how much randomness is contained in a probability distribution. The min-entropy of a distribution is a variant of the Shannon entropy which measures the amount of randomness in the worst-case as opposed to Shannon entropy which measures the expected amount of randomness in the distribution. Intuitively, a distribution with min-entropy $k$ contains $k$ random bits. A distribution $X$ is said to have min-entropy of at least $k$ bits if the probability it assigns to each element in its range is bounded above by $2^{-k}$. A distribution with min-entropy at least $k$ is called a $k$-source.

**Definition 2.** The min-entropy of a random variable $X$, taking values in a set $V$, is the quantity

$$H_{\infty}(X) \triangleq \min_{v \in V} (-\log_2 \Pr[X = v]).$$ 

### 2.4 Statistical Distance

We use statistical distance as the measure of distance between two random variables.

**Definition 3.** Let $X$ and $Y$ be random variables which both take values in a finite set $S$ with probability distributions $P_X$ and $P_Y$. The statistical distance between $X$ and $Y$ is defined as

$$\Delta [X, Y] := \frac{1}{2} \|P_X - P_Y\|_1 = \frac{1}{2} \sum_{s \in S} |P_X(s) - P_Y(s)|.$$ 

We say that $X$ and $Y$ are $\epsilon$-close if $\Delta [X, Y] \leq \epsilon$. In other words, $X$ and $Y$ are $\epsilon$-close if $|P_X(S') - P_Y(S')| \leq \epsilon$ for every event $S' \subseteq S$.

The statistical distance is the largest possible difference between the probabilities that the two probability distributions can assign to the same event. If the statistical distance between two random variables is small, then no probabilistic algorithm can distinguish between them without sampling a large amount of data. We will use the following properties of statistical distance which follow directly from the definition.
Theorem 4. Let \( X, Y \) and \( Z \) be random variables taking values in a finite set \( S \). We have

1. \( 0 \leq \Delta [X, Y] \leq 1 \)

2. \( \Delta [X, Z] \leq \Delta [X, Y] + \Delta [Y, Z] \) (triangle inequality)

In this paper, we often use statistical distance as the measure of distance between two probability distributions described by the random variables. We also use the terms statistical distance and distance in total variation interchangeably to mean the same measure.

Theorem 5 ([12]). Let \( X \) and \( Y \) be random variables which take values in a finite set \( S \) with probability distributions \( P_X \) and \( P_Y \). For every set \( S' \subseteq S \),

\[
\Delta [X, Y] \geq |P_X(S') - P_Y(S')|,
\]

and equality holds for some \( S' \subseteq S \), and in particular, for the set

\[
S' := \{ s \in S : P_X(s) < P_Y(s) \},
\]

as well as its complement.

Proof. Let us partition the set \( S \) into two disjoint subsets \( S_0 \) and \( S_1 \) as defined below.

\[
S_0 = \{ s \in S | P_X(s) < P_Y(s) \} \quad \text{and the set} \quad S_1 = \{ s \in S | P_X(s) \geq P_Y(s) \}
\]

Since \( P_X \) and \( P_Y \) are probability distributions,

\[
\sum_{s \in S_0} P_X(s) + \sum_{s \in S_1} P_X(s) = \sum_{s \in S_0} P_Y(s) + \sum_{s \in S_1} P_Y(s) = 1
\]

This implies that,

\[
\sum_{s \in S_0} P_X(s) - \sum_{s \in S_0} P_Y(s) = \sum_{s \in S_1} P_Y(s) - \sum_{s \in S_1} P_X(s)
\]

\[
\Rightarrow \sum_{s \in S_0} P_X(s) - P_Y(s) = | \sum_{s \in S_1} P_X(s) - P_Y(s) |
\]

Now,

\[
\Delta [X, Y] = \frac{1}{2} \sum_{s \in S} |P_X(s) - P_Y(s)|
\]

\[
= \frac{1}{2} \sum_{s \in S_0} |P_X(s) - P_Y(s)| + \frac{1}{2} \sum_{s \in S_1} |P_X(s) - P_Y(s)|
\]

\[
= \sum_{s \in S_0} |P_X(s) - P_Y(s)| \quad \text{or} \quad \sum_{s \in S_1} |P_X(s) - P_Y(s)|
\]

For any subset \( S' \) of \( S \)

\[
P_X(S') - P_Y(S') = P_X(S' \cap S_0) + P_X(S' \cap S_1) - (P_Y(S' \cap S_0) + P_Y(S' \cap S_1))
\]

\[
= (P_X(S' \cap S_1) - P_Y(S' \cap S_1)) - (P_Y(S' \cap S_0) - P_X(S' \cap S_0))
\]

So, we can observe that \(|P_X(S') - P_Y(S')|\) is maximized when \( S' = S_0 \) or \( S' = S_1 \), in which case it is equal to \( \Delta [X, Y] \). When \( \Delta [X, Y] \) is small, then there is no statistical test that can effectively distinguish between the distributions of \( X \) and \( Y \). \qed
**Theorem 6** ([12]). If $S$ and $T$ are finite sets, $X$ and $Y$ are random variables taking values in the set $S$ with probability distributions $P_X$ and $P_Y$, and $f : S \to T$ is a function, then $\Delta[f(X), f(Y)] \leq \Delta[X, Y]$.

**Proof.** We know from Theorem 5 that

$$
\Delta[f(X), f(Y)] = |P_{f(X)}(T') - P_{f(Y)}(T')| \quad \text{for some } T' \subseteq T
$$

$$= |P_X(f^{-1}(T')) - P_Y(f^{-1}(T'))|$$

$$\leq \Delta[X, Y].$$

where the final inequality follows again from Theorem 5.

See [12] for further discussions on statistical distance and their properties.

### 2.5 Randomness Extractors

Randomness extractors are deterministic functions that operate on arbitrary distributions with sufficient randomness and output “almost” uniformly distributed, independent random bits. Extractors require an additional input: a short seed of truly random bits as a catalyst to “extract” randomness from such distributions, i.e., the input of an extractor contains two independent sources of randomness: the actual distribution (the source) and the seed.

**Definition 4.** A $(k, \epsilon)$-extractor is a function

$$\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

such that for every distribution $X$ on $\{0,1\}^n$ with $H_\infty(X) \geq k$, the distribution $\text{Ext}(X, U_d)$ is $\epsilon$-close to the uniform distribution on $\{0,1\}^m$.

For our application, we require a stronger property from the extractor. We need the output of the extractor to remain uniform given the knowledge of the seed used. In other words, we require the extractor to extract randomness only from the source and not from the seed. A way of enforcing this condition is to demand that when the seed is concatenated to the output, the resulting distribution is still $\epsilon$-close to uniform. Such an extractor is called a strong extractor to distinguish from the non-strong extractors defined above. Non-strong Extractors guarantee to extract randomness from $k$-sources on an average seed while strong extractors guarantee to extract randomness for most seeds. In this paper, we use the term extractor to refer to a strong extractor. Extractors (strong) were first defined by Nisan and Zuckerman [2].

**Definition 5.** A $(k, \epsilon)$-strong extractor is a function

$$\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

such that for every distribution $X$ on $\{0,1\}^n$ with $H_\infty(X) \geq k$, the distribution $U_d \circ \text{Ext}(X, U_d)$ is $\epsilon$-close to the uniform distribution on $\{0,1\}^{m+d}$.

We refer to $n$ as the length of the source, $k$ as the min-entropy threshold and to $\epsilon$ as the error of the extractor, the ratio $k/n$ as the entropy rate of the source $X$ and to the ratio $m/k$ as the fraction of randomness extracted by the extractor. The entropy loss of the extractor is defined as $k + d - m$. The two inputs of the extractor have joint min-entropy of at least $k + d$ and the entropy loss measures how much of this randomness was “lost” in the extraction process. Radhakrishnan and Ta-Shma [8] showed that every non-trivial $(k, \epsilon)$-extractor cannot extract all the randomness present in its inputs and suffers an entropy loss of $\chi = 2\log(1/\epsilon) + O(1)$. For our application, we need efficient, explicit strong extractor constructions as defined below.
**Definition 6** ([11]). For functions $k(n), \epsilon(n), d(n), m(n)$ a family $\text{Ext} = \{\text{Ext}_n\}$ of functions

$$\text{Ext}_n : \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$$

is an explicit $(k,\epsilon)$-strong extractor if $\text{Ext}(x,y)$ can be computed in polynomial time in its input length $\text{poly}(n,d(n))$ and for every $n$, $\text{Ext}_n$ is a $(k(n),\epsilon(n))$-extractor.

An important property of strong extractors which makes it attractive for our application is that for any $k$-source, a $(1-\epsilon)$ fraction of the seeds extract randomness from that source. The following theorem asserts this statement formally and follows directly from the definition of a strong extractor.

**Theorem 7** ([10]). Let $\text{Ext}$ be a $(k,\epsilon)$-strong extractor, $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for every distribution $X$ on $\{0,1\}^n$ with $H_\infty(X) \geq k$, the distribution $U_d \circ \text{Ext}(X,U_d)$ is $\epsilon$-close to the uniform distribution on $\{0,1\}^{m+d}$. Let $s \in \{0,1\}^d$. Then,

$$\Pr_s[\Delta[\text{Ext}(X,s),U_m] \geq \sqrt{\epsilon}] \leq \sqrt{\epsilon}$$

*Proof.* From the definition of a $(k,\epsilon)$-strong extractor we know that $\Delta[U_d \circ \text{Ext}(X,U_d),U_{m+d}] \leq \epsilon$. By the definition of statistical distance, for $\omega \in \{0,1\}^m$ and seed $S \in \{0,1\}^d$, this can be written as

$$\frac{1}{2} \Sigma_{s,\omega} \Pr[S = s \land \text{Ext}(X,s) = \omega] - 2^{-(m+d)} \leq \epsilon$$

$$\Rightarrow \Sigma_s \Sigma_{\omega} \Pr[S = s \land \text{Ext}(X,s) = \omega] - 2^{-(m+d)} \leq 2\epsilon.$$

The expectation can then be obtained as

$$\mathbb{E}_s[\Delta[\text{Ext}(X,s),U_m]] \leq \epsilon.$$

We now invoke Markov’s inequality from Theorem 1 to conclude that

$$\Pr_s[\Delta[\text{Ext}(X,s),U_m] \geq \sqrt{\epsilon}] \leq \sqrt{\epsilon}.$$

\[\square\]

See the survey papers [11, 6, 7] for more details on extractors and their properties. In this paper, we use the explicit strong extractor construction by Raz, Reingold and Vadhan [9] which works on sources of any min-entropy on strings of length $n$. It extracts all the min-entropy using $O(\log^3 n)$ additional random seed bits while achieving an optimal entropy loss (up to an additive constant) of $\chi = 2\log(1/\epsilon) + O(1)$ bits.

**Theorem 8** (RRV Extractor [9]). For every $n, k \in \mathbb{N}$, and $\epsilon > 0$ such that $k \leq n$, there are explicit $(k,\epsilon)$-strong extractors

$$\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k-\chi}$$

with entropy loss

$$\chi = 2\log(1/\epsilon) + O(1) \text{ bits}$$

and requires seeds of length

$$d = O(\log^2 n \cdot \log(1/\epsilon) \cdot \log k) \text{ bits}.$$
2.6 Channel

The security of a steganography protocol is measured by the adversary’s ability to distinguish between “normal” and “covert” messages over a communication channel. To characterize normal communication we need to define and formalize the communication channel. We follow the standard terminology used in the literature [3, 1, 14] to define communication channels.

We let $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ denote the symbols of an alphabet and treat the channel, which will be used for data transmission, as a family of random variables $\mathcal{C} = \{C_h\}_{h \in \Sigma^*}$; each $C_h$ is supported on $\Sigma$. These channel distributions model a history-dependent notion of channel data: if $h_1, h_2, \ldots, h_t$ have been sent along the channel thus far, $C_{h_1, \ldots, h_t}$ determines the distribution of the next channel element.

We let $C_h$ denote the marginal channel distribution on a single symbol from $\Sigma$ conditioned on the history $h$. This definition of a channel differs from the typical setting where every symbol from the alphabet is drawn independently according to some fixed distribution. This definition captures the adaptive nature of the channel by making explicit the dependence between the symbols typical in real world communications.

We assume that the channel satisfies a min-entropy constraint for all histories. This assumption is important and reasonable since without this assumption it is not possible to maintain positive information content in communications. In particular, we require that $\mathcal{C}$ has min-entropy $\delta$, so that $\forall h \in \Sigma^*, H_\infty(C_h) \geq \delta$.

Observe that this definition implies that $H_\infty(C_h) \geq \delta t$.

2.7 Stegosystem

Definition 7. A one-time stegosystem consists of three probabilistic polynomial time algorithms

$$S = (SK, SE, SD)$$

where:

- $SK$ is the key generation algorithm; we write $SK(1^\nu, \log(1/\epsilon_{sec})) = k$. It takes as input, the security parameter $\epsilon_{sec}$ and the length of the message $\nu$ and produces a key $k$ of length $\kappa$. (We typically assume that $\kappa = \kappa(\nu)$ is a monotonically increasing function of $\nu$.)

- $SE$ is the embedding procedure, which can access the channel; $SE(1^\nu, k, \bar{m}, h) = c_{\text{stego}} \in \Sigma^\lambda$. It takes as input the length of the message $\nu$, the key $k$, a message $\bar{m} \in M_\nu \triangleq \{0, 1\}^\nu$ to be embedded, and the history $h$ of previously drawn covertexts. The output is the stegotext $c_{\text{stego}} \in \Sigma^\lambda$.

- $SD$ is the extraction procedure; $SD(1^\nu, k, c \in \Sigma^*) = \bar{m}$ or fail. It takes as input $1^\nu$, $k$, and some $c \in \Sigma^*$. The output is a message $m$ or the token fail.

Recall that the min-entropy of a random variable $X$, taking values in a set $V$, is the quantity

$$H_\infty(X) \triangleq \min_{v \in V} (-\log \Pr[X = v]).$$

We say that a channel $\mathcal{C}$ has min-entropy $\delta$ if for all $h \in \Sigma^*$, $H_\infty(C_h) \geq \delta$.

Definition 8 (Soundness). A stegosystem $S = (SK, SE, SD)$ is said to be $(\psi(\kappa), \delta)$-sound provided that for all channels $\mathcal{C}$ of min-entropy $\delta$,

$$\forall \bar{m} \in M_\nu, \Pr[SD(1^\nu, k, SE(1^\kappa, k, \bar{m}, h)) \neq \bar{m} | k \leftarrow SK(1^\nu, \log(1/\epsilon_{sec}))] \leq \psi(\kappa).$$

One-time stegosystem security is based on the indistinguishability between a transmission that contains a steganographically embedded message and a transmission that contains no embedded messages. An adversary $A$ against a one-time stegosystem $S = (SK, SE, SD)$ is a pair of algorithms $A = (SA_1, SA_2)$, that plays the following game, denoted $G^A(1^n)$:

1. A key $k$ is generated by $SK(1^\nu, \log(1/\epsilon_{sec})).$
2. Algorithm \( SA_1 \) receives as input the length of the message \( \nu \) and outputs a triple \( (\vec{m}^*, \theta, h_c) \in M_\nu \times \{0,1\}^* \times \Sigma^* \), where \( \theta \) is some additional information that will be passed to \( SA_2 \). \( SA_1 \) is provided access to \( \mathcal{C} \) via an oracle \( \mathcal{O}(h) \), which takes the history \( h \) as input. \( \mathcal{O}(\cdot) \), on input \( h \), returns to \( SA_1 \) an element \( c \) selected according to \( C_h \).

3. A bit \( b \) is chosen uniformly at random.
   - If \( b = 0 \) let \( c^* \leftarrow SE(1^\nu, k, m^*, h) \), so \( c^* \) is a stegotext.
   - If \( b = 1 \) let \( c^* = c_1 \circ \cdots \circ c_\lambda \), where \( \circ \) denotes string concatenation and \( c_i \leftarrow C_{h_{oc_1 \circ \cdots \circ oc_{i-1}}} \).

4. The input for \( SA_2 \) is \( 1^\nu, h_c, c^* \) and \( \theta \). \( SA_2 \) outputs a bit \( b' \). If \( b' = b \) then we say that \((SA_1, SA_2)\) succeeded and write \( G^A(1^\nu) = \text{success} \).

The advantage of the adversary \( A \) over a stegosystem \( S \) is defined as:

\[
\text{Adv}_S^A(\nu) = \left| \Pr \left[ G(1^\nu) = \text{success} \right] - \frac{1}{2} \right| .
\]

The probability includes the coin tosses of \( A \) and \( SE \), as well as the coin tosses of \( G(1^\nu) \). The (information-theoretic) insecurity of the stegosystem is defined as

\[
\text{InSec}_S(\nu) = \max_A \{ \text{Adv}_S^A(\nu) \},
\]

this maximum taken over all (time unbounded) adversaries \( A \).

**Definition 9.** (Security) We say that a stegosystem \( S = (SK, SE, SD) \) is \((\phi(\nu), \delta)\)-secure if for all channels with min-entropy \( \delta \) we have \( \text{InSec}_S(\nu) \leq \phi(\nu) \).

### 2.8 Rejection Sampling

A common method used in steganography employing a channel distribution is that of rejection sampling (cf. [1, 3]). We use a variant of rejection sampling to transmit bit vectors as opposed to a single bit. Assuming that one wishes to transmit a bit vector \( \vec{m} \in \{0,1\}^n \) and employs a random function \( f : \{0,1\}^q \times \Sigma^t \rightarrow \{0,1\}^n \) that is secret from the adversary, one performs the following “rejection sampling” process:

\[
\text{rejsam}^f_j(\vec{m}, \rho)
\]

Let \( j = 0 \)

repeat:

\[
\text{sample } \vec{c} \leftarrow C_h^t, \text{increment } j
\]

until \( f(\vec{c}) = \vec{m} \) or \( j > \rho \)

output: \( \vec{c} \)

For a given history \( h \), the procedure \( \text{rejsam}^f_j(\vec{m}, \rho) \) draws independent samples from the channel distribution \( C_h^t \) in rounds until \( f(\vec{c}) = \vec{m} \) or \( (j > \rho) \). As there are at most a total of \( \rho + 1 \) rounds, if none of the first \( \rho \) samples drawn map to the target bit vector, the sample drawn at round \( \rho + 1 \) is returned by the procedure. Here, as defined before, \( \Sigma \) denotes the output alphabet of the channel, \( h \) denotes the history of the channel data at the start of the process, and \( C_h^t \) denotes the marginal distribution on sequences of \( t \) symbols, \( \Sigma^t \), given by the channel after history \( h \). The receiver (also privy to the function \( f \)) applies the function to the received message \( \vec{c} \in \Sigma^t \) and recovers \( \vec{m} \) with a non-negligible success probability. The sender and the receiver may employ a joint state denoted by \( q \) in the above process (e.g., a counter), that need not be secret from the adversary. Note that the above process performs \( \rho + 1 \) draws from the channel with the same history. These draws are assumed to be independent. One basic property of rejection sampling that we use is:
Lemma 9 ([14]). If the function $f$ is $\epsilon$-biased on $C^t_h$ for history $h$, then for any $\rho$, $\tilde{m} \in \{0,1\}^9$:

$$\Delta \left[ \text{rejsam}^f_h(\tilde{m}, \rho), C^t_h \right] \leq \epsilon.$$  

Proof. Let us denote the samples drawn by the procedure $\text{rejsam}^f_h(\tilde{m}, \rho)$ as $\tilde{c}_i$, $i = 1, \ldots, \rho + 1$. Suppose the target bit vector $\tilde{m}$ was chosen with the probability $\Pr[f(C^t_h) = \tilde{m}]$, we first show that the output from $\text{rejsam}^f_h(\tilde{m}, \rho)$ is distributed identically to $C^t_h$. For simplicity of notation define $p^\epsilon_m \triangleq \Pr[f(\tilde{c}) = \tilde{m}]$ for $\tilde{c} \in C^t_h$. $p_c$ denotes the probability of drawing $\tilde{c}$ from the channel distribution $C^t_h$, i.e., $p_c \triangleq \Pr[\tilde{c}' \leftarrow C^t_h[\tilde{c}' = \tilde{c}].$

The probability of observing $\tilde{c} \in C^t_h$ under the $\text{rejsam}^f_h(\tilde{m}, \rho)$ procedure is then given by

$$Pr[\text{rejsam}^f_h(\tilde{m}, \rho) = \tilde{c}] = \Pr_{\tilde{c}_1 \leftarrow C^t_h}[\tilde{c}_1 = \tilde{c}] \cdot \Pr[f(\tilde{c}_1) = \tilde{m}] + \Pr_{\tilde{c}_2 \leftarrow C^t_h}[\tilde{c}_2 = \tilde{c}] \cdot \Pr[f(\tilde{c}_2) = \tilde{m}] \cdot \Pr[f(\tilde{c}_1) \neq \tilde{m}]$$

$$+ \Pr_{\tilde{c}_3 \leftarrow C^t_h}[\tilde{c}_3 = \tilde{c}] \cdot \Pr[f(\tilde{c}_3) = \tilde{m}] \cdot \Pr[f(\tilde{c}_1) \neq \tilde{m} \land f(\tilde{c}_2) \neq \tilde{m}] + \cdots$$

$$= p_c p^\epsilon_m + p_c p^\epsilon_m (1 - p^\epsilon_m) + p_c p^\epsilon_m (1 - p^\epsilon_m)^2 + p_c p^\epsilon_m (1 - p^\epsilon_m)^3$$

$$+ \cdots + p_c p^\epsilon_m (1 - p^\epsilon_m)^{\rho-1} + p_c (1 - p^\epsilon_m)$$

$$= p_c p^\epsilon_m \left( 1 + (1 - p^\epsilon_m) + (1 - p^\epsilon_m)^2 + (1 - p^\epsilon_m)^3 + \cdots + (1 - p^\epsilon_m)^{\rho-1} \right) + p_c (1 - p^\epsilon_m)^\rho$$

$$= p_c p^\epsilon_m \left( \frac{1 - (1 - p^\epsilon_m)^\rho}{p^\epsilon_m} \right) + p_c (1 - p^\epsilon_m)^\rho$$

But since $\Delta[\tilde{m}, U_9] \leq \epsilon$, it must be the case that $\Delta \left[ \text{rejsam}^f_h(\tilde{m}, \rho), C^t_h \right] \leq \epsilon$ by Theorem 6.  

3 The construction

In this section we outline our construction of a one-time stegosystem as an interaction between Alice (the sender) and Bob (the receiver). Alice and Bob wish to communicate over a channel with distribution $\mathcal{E}$. For simplicity, we assume that the support of $\mathcal{E}$ is of size $|\Sigma| = 2^b$. We also assume that $\mathcal{E}$ has min-entropy $\delta$, so that $\forall h \in \Sigma^*, H_\infty(C_h) \geq \delta$ and by the additive property of min-entropy, $H_\infty(C_h^t) \geq \delta t$, i.e., $C_h$ is a $\delta$-source and $C_h^t$ is a $\delta t$-source.

3.1 A one-time stegosystem

Fix an alphabet $\Sigma$ for the channel, choose a message $\tilde{m}' \in \{0,1\}^\nu$ and the security parameter $\epsilon_{sec}$. Our stegosystem uses the $RRV$ strong-extractor construction as described in Theorem 8 which extracts randomness from the marginal channel distribution $C^t_h$ supported on $\{0,1\}^{3b}$ by rejection sampling as described in Section 2.8.

Alice and Bob agree on the following:

**Extractor Construction.** Alice and Bob agree to use the explicit $RRV$ strong-extractor construction as described in Theorem 8. They use a shared seed $s \in R \{0,1\}^d$. The notation $E_s \triangleq E_s(\cdot)$ stands for the extractor $E(\cdot, \cdot)$ used with the seed $s$ i.e., $E(\cdot, s)$.

**One-Time Pad.** Alice and Bob use a shared secret key $k^{OTP} \in R \{0,1\}^\nu$ to randomize their message.

Key generation consists of selecting the seed $s \in R \{0,1\}^d$ and the one-time pad secret key $k^{OTP} \in R \{0,1\}^\nu$. For our protocol, the shared seed need not be secret. The encoding procedure accepts an input message and outputs stego text of length $\lambda$. We will analyze the stegosystem below in terms of arbitrary parameters $t, d, \lambda, \epsilon$ and $\epsilon_{sec}$ relegating discussion of how these parameters determine the overall efficiency of the system to Section 3.4.
Alice and Bob then communicate using the algorithm $SE$ for embedding and $SD$ for extracting described in Figure 1. In $SE$, after applying the one-time pad, we use $\text{rejsam}_{\nu}^{E'}(\vec{m}_i, \rho)$ to obtain an element $\vec{c}_i$ of the channel for each block $\vec{m}_i$ of the message. Here, the history $h$ represents the current history at the time of rejection sampling which gets updated after sampling. The resulting stegotext $\vec{c}_1\vec{c}_2\ldots \vec{c}_{\ceil{\nu/c\log \nu}}$ is denoted $c_{\text{stego}}$. In $SD$ the received stegotext is parsed block by block by evaluating the extractor using seed $s$; this results in a message block. After performing this for each received block, a message of size $\nu$ is received, which is subjected to the one-time pad decoding to obtain the original message.

\begin{table}[h]
\centering
\begin{tabular}{ |c|c| }
\hline
\textbf{PROCEDURE $SE(k, \vec{m}', h)$:} & \textbf{PROCEDURE $SD(k, c_{\text{stego}})$:} \\
\hline Input: Key $k = (k_{\text{OTP}}, s)$; hidden text $\vec{m}' \in \{0,1\}^\nu$, history $h \in \Sigma^*$ & Input: Key $k = (k_{\text{OTP}}, s)$ stegotext $c_{\text{stego}}$ \\
let $\vec{m} = k_{\text{OTP}} \oplus \vec{m}'$ & parse $c_{\text{stego}}$ as $c = \vec{c}_1 \circ \vec{c}_2 \ldots \vec{c}_{\ceil{\nu/c\log \nu}}$ \\
parse $\vec{m}$ as $\vec{m} = \vec{m}_1 \vec{m}_2 \ldots \vec{m}_{\ceil{\nu/c\log \nu}}$ & for $i = 1$ to $\ceil{\nu/c\log \nu}$ do \\
for $i = 1$ to $\ceil{\nu/c\log \nu}$ \{ & set $\vec{m}_i = E_s(\vec{c}_i)$ \\
$\vec{c}_i = \text{rejsam}_{\nu}^{E'}(\vec{m}_i, \rho)$ & \} \\
set $h \leftarrow h \circ \vec{c}_i$ & set $\vec{m} = \vec{m}_1 \vec{m}_2 \ldots \vec{m}_{\ceil{\nu/c\log \nu}}$ \\
\} & Output: $\vec{m} = k_{\text{OTP}} \oplus \vec{m}'$ \\
Output: $c_{\text{stego}} = \vec{c}_1\vec{c}_2\ldots \vec{c}_{\ceil{\nu/c\log \nu}} \in \Sigma^\lambda$ & \\
\hline
\end{tabular}
\end{table}

Figure 1: Encryption and Decryption algorithms for the one-time stegosystem of 3.1.

The detailed security and correctness analysis follow in the next two sections.

### 3.2 Security

In this section we argue about the security of our one-time stegosystem. We wish to quantify the security of our stegosystem by the statistical distance between the “normal” and “covert” message distributions over the communication channel. First, by Lemma 9, observe that if the function $f$ is $\epsilon$-biased on $C_h^\lambda$ for history $h$, then for any $\rho, \vec{m} \in R \{0,1\}^\nu, \Delta[\text{rejsam}_h^E(\vec{m}, \rho), C_h^\lambda] \leq \epsilon$. It follows from the definition of a strong-extractor, Theorem 7, that for an uniformly chosen seed $s \in R \{0,1\}^d$, $\Pr_s[\Delta[\text{Ext}(X,s), U_m] \geq \sqrt{\epsilon}] \leq \sqrt{\epsilon}$. So,

\[
\Delta[\text{rejsam}_h^E(\vec{m}, \rho), C_h^\lambda] \leq (1 - \sqrt{\epsilon}) \cdot \sqrt{\epsilon} + \sqrt{\epsilon} \cdot 1 = 2\sqrt{\epsilon} - \epsilon \leq 2\sqrt{\epsilon}.
\]

Suppose in our stegosystem construction, [Figure 1], we had used an independent and uniformly chosen seed $s_i \in \{0,1\}^d$ for each message block $i = 1,2,\ldots, \ceil{\nu/c\log \nu}$, the statistical distance between the natural channel distribution and the output of the encoding procedure $SE$ can be given by

\[
\Delta[SE(k, \vec{m}', h), C_h^\lambda] \leq 2\sqrt{\epsilon}[\nu/c\log \nu] \text{ by the triangle inequality.}
\]

Next, we present an upper bound on the statistical distance between the “natural” channel distribution and the output of the encoding procedure $SE$ when using a single seed $s \in R \{0,1\}^d$ over all the message blocks as in our construction.

We need the following technical lemmas to prove the results of this section.

**Lemma 10.** Consider random variables $X, X' \in \mathcal{X}$ and $Y_X, Y'_X \in \mathcal{Y}$ for each $x \in \mathcal{X}$. Then,

\[
\Delta[(X, Y_X), (X', Y'_X)] \leq \Delta[X, X'] + \Delta[Y_X, Y'_X].
\]
Proof. For \( x \in \mathcal{X} \) denote \( \Pr[X = x] \) by \( P_x \) and \( \Pr[Y_x = y] \) by \( P_y|x \). Then we get

\[
\Delta \left[ (X, Y_X), (X', Y'_X) \right] = \frac{1}{2} \sum_{x,y \in \mathcal{Y}} \left| P_x \cdot P_y|x - P'_x \cdot P'_y|x \right|
\]

\[
\leq \frac{1}{2} \sum_{x,y} \left| P_x \cdot P_y|x - P'_x \right| + \frac{1}{2} \sum_{x,y} \left| P'_x \cdot P_y|x - P'_y|x \right| \quad \text{(triangle inequality)}
\]

\[
= \frac{1}{2} \sum_{x,y} P_y|x \cdot |P_x - P'_x| + \frac{1}{2} \sum_{x,y} P'_x \cdot |P_y|x - P'_y|x
\]

\[
= \Delta [X, X'] + \Delta [Y_X, Y'_X].
\]

\( \square \)

Lemma 11. Consider random variables \( X, X' \in \mathcal{X} \) and \( Y_X, Y'_X, \in \mathcal{Y} \) for each \( x \in \mathcal{X} \). Then for any constant \( \beta > 0 \),

\[
\Delta [Y_X, Y'_X] \leq \Pr_X \left[ \Delta [Y_X, Y'_X] > \beta \right] \cdot 1 + \Pr_X \left[ \Delta [Y_X, Y'_X] \leq \beta \right] \cdot \beta.
\]

Theorem 12. For a message \( \vec{m}' \in \{0,1\}^\nu \), the insecurity of the stegosystem \( S \) of Section 3.1 is bound by \( \ell \cdot (\sqrt{\epsilon_{\text{sec}}} + 2\sqrt{\epsilon_{\text{sec}}}) \), i.e., \( \text{InSec}(S, \nu) \leq \ell \cdot (\sqrt{\epsilon_{\text{sec}}} + 2\sqrt{\epsilon_{\text{sec}}}) \), where \( \epsilon_{\text{sec}} \) is the security parameter and \( \ell = [\nu/c \log \nu] \) for some constant \( c \).

Proof. We start the encoding procedure \( SE \) with history \( h \) which embeds message blocks into the channel adaptively using rejection sampling. We want to show that the statistical distance between the output of \( SE \) and the natural channel distribution is given by

\[
\Delta \left[ SE(k, \vec{m}', h), C_h \right] \leq \ell \cdot (\sqrt{\epsilon_{\text{sec}}} + 2\sqrt{\epsilon_{\text{sec}}})
\]

where \( \ell = [\nu/c \log \nu] \) for some constant \( c \) and \( \lambda \) is the length of the output by procedure \( SE \).

First, we define some notation to capture the operation of the procedure \( SE \). Let \( C_\varphi \) denote the channel distribution \( C_h \) at depth 0. Let \( C_1 \) denote the distribution at depth 1 that results by sampling \( C_1 \leftarrow C_\varphi \); \( C_2 \) denotes the distribution at depth 2 that results by sampling \( C_2 \leftarrow C_1 \). Similarly, let \( C_\tau \) denote the channel distribution at depth \( \tau \) that results by sampling \( C_{\tau-1} \leftarrow C_\tau \). We define the random variables obtained by rejection sampling in the same fashion. Let us now formally define these families of random variables \( C_i, R_i \) at depth \( i \), \( i = 0, 1, \cdots, \ell \). Here, \( \vec{m} = k^{\text{OTP}} \oplus \vec{m}' = \vec{m}_1 \circ \vec{m}_2 \circ \cdots \circ \vec{m}_\ell \) and \( |\vec{m}_i| = c \log \nu \) for \( i = 1, \cdots, \ell \).

\[
C_\varphi \triangleq C_h
\]

\[
C_1 \triangleq C_{h \circ c_1} \text{ where } c_1 \leftarrow C_h
\]

\[
C_2 \triangleq C_{h \circ c_2 \circ c_1} \text{ where } c_2 \leftarrow C_{h \circ c_1}
\]

\[
\vdots \quad \vdots
\]

\[
C_\tau \triangleq C_{h \circ c_\tau \circ \cdots \circ c_{\tau-1}} \text{ where } c_\tau \leftarrow C_{h \circ c_{\tau-2} \circ \cdots \circ c_{\tau-1}}
\]

and

\[
R_\varphi \triangleq C_h
\]

\[
R_1 \triangleq C_{h \circ c_1} \text{ where } c_1 \leftarrow \text{rejsam}_{\tau}^{E_{\varphi}(\cdot)}(\vec{m}_1, \rho)
\]

\[
R_2 \triangleq C_{h \circ c_2 \circ c_1} \text{ where } c_2 \leftarrow \text{rejsam}_{\tau}^{E_{\varphi}(\cdot)}(\vec{m}_2, \rho)
\]

\[
\vdots \quad \vdots
\]

\[
R_\tau \triangleq C_{h \circ c_\tau \circ \cdots \circ c_{\tau-1}} \text{ where } c_\tau \leftarrow \text{rejsam}_{\tau}^{E_{\varphi}(\cdot)}(\vec{m}_\tau, \rho).
\]

11
Now, pick a seed \( s \in_R \{0,1\}^d \). First, we show that at each depth \( \tau \), \( \tau = 1, \ldots, \ell \), the probability mass of distributions for which the extractor coupled with the seed \( s \) yields an \( \sqrt{\varepsilon} \)-biased function is large.

We say that a distribution \( C_{c_1,c_2,\ldots,c_\ell} \) is \( (s,\sqrt{\varepsilon_{sec}}) \)-good if \( E_s \triangleq E_s() \triangleq E(\cdot,s) \) is \( \sqrt{\varepsilon_{sec}} \)-biased on \( C \). Otherwise we say that the distribution \( C \) is \( (s,\sqrt{\varepsilon_{sec}}) \)-bad.

Let us define the following sets:

\[
\tau \quad \text{G}_s^\tau = \{ (c_1,c_2,\ldots,c_\tau) | C_{c_1,c_2,\ldots,c_\tau} \text{ is } (s,\sqrt{\varepsilon_{sec}}) \text{-good} \}
\]

and

\[
\tau \quad \text{B}_s^\tau = \{ (c_1,c_2,\ldots,c_\tau) | C_{c_1,c_2,\ldots,c_\tau} \text{ is } (s,\sqrt{\varepsilon_{sec}}) \text{-bad} \}.
\]

\( \tau \text{G}_s^\tau \), \( \tau \text{B}_s^\tau \) denote the collection of \( (s,\sqrt{\varepsilon_{sec}}) \)-good and \( (s,\sqrt{\varepsilon_{sec}}) \)-bad distributions at depth \( \tau \) respectively.

The basic property of a strong-extractor construction that we use here is that for any distribution \( C \) with the right min-entropy, the probability over the choice of the seed \( s \), \( \Pr_{s}[C] \) is \( (s,\sqrt{\varepsilon_{sec}}) \)-good \( \geq 1 - \sqrt{\varepsilon_{sec}} \).

By Theorem 7. This implies that \( \Pr_{s}[C] \) is \( (s,\sqrt{\varepsilon_{sec}}) \)-bad \( \leq \sqrt{\varepsilon_{sec}} \).

Define the quantity \( \mu (\tau \text{G}_s^\tau) \triangleq \sum_{\bar{c} \in \tau \text{B}_s^\tau} \Pr[\text{Observing } \bar{c} \text{ under the natural channel distribution}] = \sum_{\bar{c} \in \tau \text{B}_s^\tau} \Pr_{C_s}[\bar{c}] \) with the appropriate history \( h \). This quantity \( \mu (\tau \text{G}_s^\tau) \) represents the probability mass of the set \( \tau \text{B}_s^\tau \). Define

\[
\mu (\tau \text{G}_s^\tau) = \sum_{\bar{c} \in \tau \text{G}_s^\tau} \Pr_{C_s}[\bar{c}].
\]

Define the indicator random variable

\[
X_s^\tau = \begin{cases} 
1 & \text{if } \tau \text{G}_s^\tau \text{ is } (s,\sqrt{\varepsilon_{sec}})\text{-good}, \\
0 & \text{otherwise}.
\end{cases}
\]

The expected value \( \mathbb{E}[X_s^\tau] = 1 \cdot \Pr[\tau \text{G}_s^\tau \text{ is } (s,\sqrt{\varepsilon_{sec}})\text{-good}] \geq 1 - \sqrt{\varepsilon_{sec}} \).

Now, notice that the expected mass of \( \tau \text{G}_s^\tau \) over the choice of randomly picking the seed \( s \) is given by,

\[
\mathbb{E}_{s}[\mu (\tau \text{G}_s^\tau)] = \mathbb{E}_{s}[\sum_{\bar{c} \in \tau \text{G}_s^\tau} \Pr_{C_s}[\bar{c}]] = \sum_{\bar{c} \in \tau \text{G}_s^\tau} \Pr_{C_s}[\bar{c}] \cdot \mathbb{E}[X_s^\tau] \geq 1 - \sqrt{\varepsilon_{sec}}
\]

since \( \sum_{\bar{c} \in \tau \text{G}_s^\tau} \Pr_{C_s}[\bar{c}] = 1 \).

Consequently,

\[
\mathbb{E}_{s}[\mu (\tau \text{B}_s^\tau)] \leq \sqrt{\varepsilon_{sec}}.
\]

We now want to compute the probability that this expected value is small. Using Markov’s inequality from Theorem 1 we get

\[
\Pr_{s}[\mu (\tau \text{B}_s^\tau) \geq \alpha] \leq \mathbb{E}_{s}[\mu (\tau \text{B}_s^\tau)]/\alpha.
\]

When \( \alpha = \sqrt{\varepsilon_{sec}} \), we get

\[
\Pr_{s}[\mu (\tau \text{B}_s^\tau) \geq \sqrt{\varepsilon_{sec}}] \leq \sqrt{\varepsilon_{sec}}.
\]

By Boole’s inequality from Theorem 3 we get

\[
\Pr_{s}\left[ \exists \tau \ | \mu (\tau \text{B}_s^\tau) \geq \sqrt{\varepsilon_{sec}} \right] \leq \ell \sqrt{\varepsilon_{sec}}.
\]

So,

\[
\Pr_{s}\left[ \exists \tau \ | \mu (\tau \text{G}_s^\tau) \geq 1 - \sqrt{\varepsilon_{sec}} \right] \geq 1 - \ell \sqrt{\varepsilon_{sec}}
\]

where \( \ell = \lceil \nu/c \log \nu \rceil \), the number of message blocks.
So, we can see that if we pick a seed $s$ uniformly at random, with probability $1 - \ell \sqrt{\epsilon_{sec}}$ in the choice of $s$, $\forall \tau$, $\mu(B'_\tau)\leq \tau$ is small which implies that $\mu(G'_\tau) \geq 1 - \sqrt{\epsilon_{sec}}$. Here $\tau = 1, 2, \ldots, \ell$.

We say that a seed $s$ is good if $\forall \tau$, $\mu(G'_\tau) \geq 1 - \sqrt{\epsilon_{sec}}$. From the discussion above we know that

$$\Pr_{s \in R\{0,1\}^d}[s\text{ is good}] \geq 1 - \ell \sqrt{\epsilon_{sec}}.$$ 

Now, fix a good seed $s$. We will now prove that for a good seed $s$,

$$\Delta[(C_1, C_2, \ldots, C_{\ell}),(R_1, R_2, \ldots, R_{\ell})] \leq \ell \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}).$$

We prove this by induction on $\ell$, the number of message blocks.

**Base Case:** for $\tau = 1$: $\Delta[C_1, R_1] \leq \epsilon_{sec} \leq \sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}$.

**Inductive hypothesis:** $\Delta[(C_1, C_2, \ldots, C_{\tau}), (R_1, R_2, \ldots, R_{\tau})] \leq \tau \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}})$. To show:

$$\Delta[(C_1, C_2, \ldots, C_{\tau+1}), (R_1, R_2, \ldots, R_{\tau+1})] \leq (\tau + 1) \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}).$$

Observe that,

$$\Delta[(C_1, C_2, \ldots, C_{\tau+1}), (R_1, R_2, \ldots, R_{\tau+1})] \leq \Delta[(C_1, C_2, \ldots, C_{\tau}), (R_1, R_2, \ldots, R_{\tau})] + \Delta[(C_1, C_2, \ldots, C_{\tau}, C_{\tau+1}), (R_1, R_2, \ldots, R_{\tau+1})]$$

by Lemma 10

$$\leq \tau \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}) + \Delta[(C_1, C_2, \ldots, C_{\tau}, C_{\tau+1}), (C_1, C_2, \ldots, C_{\tau}, R_{\tau+1})]$$

$$\leq \tau \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}) + (1 - \sqrt{\epsilon_{sec}}) \cdot \sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}} \cdot 1$$

by Lemma 11

$$\leq \tau \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}) + (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}) \leq (\tau + 1) \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}).$$

Hence we can conclude that for a good seed $s$,

$$\Delta[(C_1, C_2, \ldots, C_{\ell}), (R_1, R_2, \ldots, R_{\ell})] \leq \ell \cdot (\sqrt{\epsilon_{sec}} + \sqrt{\epsilon_{sec}}).$$

So, the statistical distance is now given by,

$$\Delta[(C_1, C_2, \ldots, C_{\ell}), (R_1, R_2, \ldots, R_{\ell})] = \Delta[(C_1, C_2, \ldots, C_{\ell}), (R_1, R_2, \ldots, R_{\ell})]|_{s \text{ is good}} \cdot \Pr[s \text{ is good}] + \Delta[(C_1, C_2, \ldots, C_{\ell}), (R_1, R_2, \ldots, R_{\ell})]|_{s \text{ is not good}} \cdot \Pr[s \text{ is not good}]
\leq \ell \cdot \sqrt{\epsilon_{sec}} \cdot \sqrt{\epsilon_{sec}} \cdot (1 - \sqrt{\epsilon_{sec}}) + 1 \cdot (\sqrt{\epsilon_{sec}}) + 1 \cdot (\sqrt{\epsilon_{sec}})
\leq \ell \cdot (\sqrt{\epsilon_{sec}} + 2 \sqrt{\epsilon_{sec}}).$$

Thus,

$$\Delta[SE(k, \hat{m}', h), C_{h}] \leq \ell \cdot (\sqrt{\epsilon_{sec}} + 2 \sqrt{\epsilon_{sec}})$$

where $\ell = \lceil \nu/c \log \nu \rceil$ for some constant $c$.

and the theorem follows by the definition of insecurity. □

13
3.3 Correctness

In this section we obtain an upper bound on the soundness of our stegosystem. We focus on the mapping between \( \{0,1\}^\nu \) and \( \Sigma^k \) determined by the \( SE \) procedure of the one-time stegosystem. We would like to bound the probability of the stego decoding procedure’s inability to faithfully recover the encoded message. Following the definition of soundness from Section 2.7 we seek to upper bound this probability of failure:

\[
\Pr[SD(1^\nu, k, SE(1^\kappa, k, \vec{m}, h)) \neq \vec{m} \mid k \leftarrow SK(1^\nu, \log(1/\epsilon_{sec}))], \forall \vec{m} \in M_\nu.
\]

For simplicity of notation, let \( F \) be the event that Bob is unable to correctly decode the message sent from Alice, i.e., \( SD(1^\nu, k, SE(1^\kappa, k, \vec{m}, h)) \neq \vec{m} \mid k \leftarrow SK(1^\nu, \log(1/\epsilon_{sec})) \). Since \( \vec{m} = \vec{m}_1 \circ \vec{m}_2 \circ \cdots \circ \vec{m}_\ell \) and \( |\vec{m}_i| = c \log \nu \) for \( i = 1, \cdots, \ell \), let us first estimate the probability of failure for one message block \( \vec{m}_i \) denoted by \( F' \). We reuse the notations and definitions introduced in the security proof of the section above.

Recall that a seed \( s \) is good if \( \forall \tau \mu(G_s^{\tau}) \geq 1 - \sqrt[4]{\epsilon_{sec}} \). As shown before, the probability of seed \( s \) to be good is given by \( \Pr_s[\forall \tau \mid \mu(G_s^{\tau}) \geq 1 - \sqrt[4]{\epsilon}] \leq 1 - \ell \sqrt[4]{\epsilon} \). This gives us

\[
\Pr[F] = \ell \cdot (\Pr[F' \mid s \text{ is good}] \cdot \Pr[s \text{ is good}] + \Pr[F' \mid s \text{ is not good}] \cdot \Pr[s \text{ is not good}])
\]

\[
\leq \ell \cdot (\Pr[F' \mid s \text{ is good}] \cdot 1 + 1 \cdot (\ell \sqrt[4]{\epsilon_{sec}}))
\]

where \( \epsilon_{sec} = \frac{1}{4 \cdot 2^c |m_i|} \) and \( \rho = 2 \cdot |m_i| \cdot 2^c |m_i| \) we get

\[
\Pr[F' \mid s \text{ is good}] \leq (1 - \sqrt[4]{\epsilon_{sec}}) \cdot \left(1 - \frac{1}{2^{c |m_i|}} + \sqrt{\epsilon_{sec}} \right)^\rho + \sqrt[4]{\epsilon_{sec}} \cdot 1
\]

where \( \rho \) is the bound on the number of iterations performed by the rejection sampling procedure. When \( \epsilon_{sec} = \frac{1}{4 \cdot 2^c |m_i|} \) and \( \rho = 2 \cdot |m_i| \cdot 2^c |m_i| \) we get

\[
\Pr[F' \mid s \text{ is good}] \leq (1 - \sqrt[4]{\epsilon_{sec}}) \cdot \frac{1}{2^{c |m_i|}} + \sqrt{\epsilon_{sec}}
\]

In our construction, we have each \( |m_i| = c \log \nu \). So, \( \epsilon_{sec} = \frac{1}{4 \cdot 2^c |m_i|} = \frac{1}{4 \cdot \nu^c} \) and \( \rho = 2 \cdot c \log \nu \cdot \nu^c \). Here \( \ell = \nu/c \log \nu \).

Thus we can reduce \( \Pr[F] \) as

\[
\Pr[F] \leq \ell \cdot (\Pr[F' \mid s \text{ is good}] \cdot 1 + 1 \cdot (\ell \sqrt[4]{\epsilon_{sec}}))
\]

\[
\leq \ell \cdot \left( (1 - \sqrt[4]{\epsilon_{sec}}) \cdot \frac{1}{2^{c |m_i|}} + \sqrt{\epsilon_{sec}} \right)
\]

\[
\leq \frac{\ell}{\nu^c} + \frac{\ell}{\sqrt{2^c \nu^c}} \cdot \frac{\nu^2}{\sqrt{2^c \nu^c}}
\]

\[
= \frac{\nu}{c \log \nu \cdot \nu^c} + \frac{\nu}{c \log \nu \cdot 2^c \nu^c} + \frac{\nu^2}{c^2 \log^2 \nu \cdot 2^c \nu^c}
\]

\[
= \frac{1}{4 \nu^3 \log \nu} + \frac{1}{4 \sqrt{2} \nu \log \nu} + \frac{1}{16 \sqrt{2} \log^2 \nu}
\]

if we choose the constant \( c = 4 \)

We record the following lemma which follows from the discussion above.

**Lemma 13.** With \( SE \) and \( SD \) described as above, the probability that a message \( \vec{m} \) of length \( \nu \) is recovered from the stegosystem is at least

\[
1 - \left( \frac{1}{4 \nu^3 \log \nu} + \frac{1}{4 \sqrt{2} \nu \log \nu} + \frac{1}{16 \sqrt{2} \log^2 \nu} \right)
\]
3.4 Putting it all together

The objective of this section is to integrate the results of the paper. We first show that for a perfectly secure, perfectly sound steganographic protocol, we need at least $\nu$ bits of min-entropy in the communication channel to embed a message $m \in R \{0, 1\}^\nu$. This analysis yields a lower bound on the length of the stegotext transmitted. Next, we show that for such a steganography protocol, we also need the length of the shared secret key $k$ to be at least $\nu$ bits. This gives us a lower bound on the number of random bits used by a steganographic protocol.

Consider the following experiment: Fix a family of channel distributions. Now, pick a message $m \in R \{0, 1\}^\nu$ and fix the shared secret $k$, independent of the message $m$. Finally, fix the encoding $Enc(k, m)$ and the deterministic decoding $Dec(k, Enc(m))$ algorithm. Alice encodes the message using the encoding procedure and transmits $Enc(k, m)$ to Bob who decodes the message with no errors as the stegosystem is perfectly sound. Let us now show that the random variable $Enc(k, m)$ has $\nu$ bits of min-entropy. As the encoding function is a probabilistic function the random variable $Enc(k, m)$ depends on the message $m$ and the channel distribution. Notice that since the decoding procedure is a deterministic procedure, the distribution output by the decoding algorithm has the same min-entropy as that of its input distribution.

Now, Bob recovers $m$ from $Enc(k, m)$ without any errors. Since there are $\nu$ bits of min-entropy in the message source $m \in R \{0, 1\}^\nu$, it follows that there are $\nu$ bits of min-entropy in $Enc(k, m)$ as well. Since the stegosystem is perfectly secure, the distribution $Enc(k, m)$ is identical to the channel distribution. This would imply that the channel also has $n$ bits of min-entropy. This establishes a lower bound on the communication complexity of a steganographic protocol. Suppose we have a channel distribution such that for all valid history, the min-entropy of the channel is $\delta$, the above discussion shows that to transmit a message $m \in R \{0, 1\}^\nu$, the length of the stegotext transmitted is at least $\nu/\delta$ symbols.

Let us now focus on the number of random bits in the shared secret $k$. As the stego system is sound, the decoding procedure reconstructs the message $m$ perfectly. Furthermore, since this is a perfectly secure stegosystem, the random variables $m$ and $Enc(k, m)$ are independent. The mutual information between $m$ and $Enc(k, m)$ is zero. But, the mutual information between $m$ and the random variable pair $(Enc(k, m), k)$ is $\nu$ as Bob is able to recover $m$ from the pair $(Enc(k, m), k)$. It follows therefore that $k$ has $\nu$ bits of min-entropy, i.e., $k$ can be chosen uniformly from $\{0, 1\}^\nu$.

If we let $t = \delta^{-1} \cdot (c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1))$ for some constant $c$, the channel distribution $C^t_h$ supported on $\{0, 1\}^{\delta^{-1} \cdot (c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1))}$ has a min-entropy of at least $k = c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1)$. To put this all together, the RRV strong-extractor is a function

$$Ext : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^{k-\Delta}$$

where

$$n = \delta^{-1} \cdot (c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1)) \cdot b$$
$$d = O(\log^2 (\delta^{-1} \cdot (c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1)) \cdot b) \cdot \log (1/\epsilon_{sec}) \cdot \log k)$$
$$k = c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1)$$
$$\Delta = 2 \log (1/\epsilon_{sec}) + O(1) \text{ and }$$
$$k - \Delta = c \log \nu$$

This implies that our stegotext is of length $\frac{\nu}{c \log \nu} \cdot \delta^{-1} \cdot (c \log \nu + 2 \log (1/\epsilon_{sec}) + O(1)) \cdot b$ bits to embed $\nu$ bits of message. Alice and Bob share $\nu$ secret bits. They also use a public seed for the extractor of length $d$ bits.

4 A provably secure stegosystem for longer messages

In this section we show how to apply the “one-time” stegosystem of Section 3.1 together with a pseudorandom number generator so that longer messages can be transmitted.
Definition 10. Let $U_1$ denote the uniform distribution over $\{0,1\}^1$. A polynomial time deterministic program $G$ is a pseudorandom generator (PRG) if the following conditions are satisfied:

**Variable output** For all seeds $x \in \{0,1\}^*$ and $y \in \mathbb{N}$, $|G(x, 1^y)| = y$ and, furthermore, $G(x, 1^y)$ is a prefix of $G(x, 1^{y+1})$.

**Pseudorandomness** For every polynomial $p$ the set of random variables $\{G(U_1, 1^{p(i)})\}_{i \in \mathbb{N}}$ is computationally indistinguishable from the uniform distribution $U_{p(i)}$.

Note that there is a procedure $G'$ that if $z = G(x, 1^y)$ it holds that $G(x, 1^{y+y'}) = G'(x, z, 1^y)$ (i.e., if one maintains $z$, one can extract the $y'$ bits that follow the first $y$ bits without starting from the beginning). For a PRG $G$, if $A$ is some statistical test, then we define the advantage of $A$ over the PRG as follows:

$$\text{Adv}^A_G(l) = \left| \Pr_{i \sim G(U_1, 1^{p(i)})}[A(\hat{l}) = 1] - \Pr_{i \sim U_{p(i)}}[A(\hat{l}) = 1] \right|$$

The insecurity of the PRNG $G$ is then defined

$$\text{InSec}^G_{PRG}(l) = \max_A \{ \text{Adv}^A_G(l) \}.$$  

Note that typically in PRGs there is a procedure $G'$ as well as the process $G(x, 1^y)$ produces some auxiliary data $\text{aux}_y$ of small length so that the rightmost $y'$ bits of $G(x, 1^{y+y'})$ may be sampled directly as $G'(x, 1^y, \text{aux}_y)$. Consider now the following stegosystem $S' = (SE', SD')$ that can be used for arbitrary many and long messages and employs a PRG $G$ and the one-time stegosystem $(SK, SE, SD)$ of Section 3.1. The two players Alice and Bob, share a key of length $l$ denoted by $x$. They also maintain a state $N$ that holds the number of bits that have been transmitted already as well the auxiliary information $\text{aux}_N$ (initially empty). The function $SE'$ is given input $N, \text{aux}_N, x, m \in \{0,1\}^n$ where $m$ is the message to be transmitted. $SE'$ in turn employs the PRG $G$ to extract a number of bits $\kappa$ as follows $k = G'(x, 1^\kappa, \text{aux}_N)$. The length $\kappa$ is selected to match the number of key bits that are required to transmit the message $m$ using the one-time stegosystem of section 3.1. Once the key $k$ is produced by the PRG the procedure $SE'$ invokes the one-time stegosystem on input $k, m, h$. After the transmission is completed the history $h$, the count $N$, as well as the auxiliary PRG information $\text{aux}_N$ are updated accordingly. The function $SD'$ is defined in a straightforward way based on $SD$.

Theorem 14. The stegosystem $S' = (SE', SD')$ is provably secure in the model of [3] (universally steganographically secret against chosen hiddeentext attacks); in particular

$$\text{InSec}^SS_{SE'}(t, q, l) \leq \text{InSec}^{PRG}(t + \gamma(\ell(l)), \ell(l) + \text{polylog}(l))$$

(where $t$ is the time required by the adversary, $q$ is the number of chosen hiddentext queries it makes, $l$ is the total number of bits across all queries and $\gamma(v)$ is the time required to simulate the $SE'$ oracle for $v$ bits).

References

[1] Christian Cachin. An information-theoretic model for steganography. In *Information Hiding*, pages 306–318, 1998.

[2] Noam Nisan David and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52:43–52, 1993.

[3] Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In *CRYPTO*, pages 77–92, 2002.

[4] Aggelos Kiayias, Yona Raekow, Alexander Russell, and Narasimha Shashidhar. Efficient steganography with provable security guarantees. *Preprint, arXiv:0909.3658*, September 2009.
[5] Thomas Mittelholzer. An information-theoretic approach to steganography and watermarking. In Information Hiding, pages 1–16, 1999.

[6] N. Nisan. Extracting randomness: How and why: A survey. In Proceedings of the 11th Annual IEEE Conference on Computational Complexity, pages 44–58. Citeseer, 1996.

[7] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. Journal of Computer and System Sciences, 58(1):148–173, 1999.

[8] Jaikumar Radhakrishnan and Amnon Ta-shma. Bounds for dispersers, extractors, and depth-two. SIAM Journal on Discrete Mathematics, 13:2000, 2000.

[9] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in trevisan’s extractors. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pages 149–158, 1999.

[10] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. In Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, pages 22–31, 2000.

[11] R. Shaltiel. Recent developments in explicit constructions of extractors. Bulletin of the EATCS, 77:67–95, 2002.

[12] Victor Shoup. A computational introduction to number theory and algebra. Cambridge University Press, New York, NY, USA, 2005.

[13] Gustavus J. Simmons. The prisoners’ problem and the subliminal channel. In CRYPTO, pages 51–67, 1983.

[14] Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In Advances in Cryptology, Proceedings of Eurocrypt 04, pages 323–341. Springer-Verlag, 2004.

[15] Jan Zöllner, Hannes Federrath, Herbert Klimant, Andreas Pfitzmann, Rudi Piotraschke, Andreas Westfeld, Guntram Wicke, and Gritta Wolf. Modeling the security of steganographic systems. In Information Hiding, pages 344–354, 1998.