# Upper and Lower Bounds on Black-Box Steganography

## Extended Abstract


Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell

Boston University Computer Science
111 Cummington St.
Boston MA 02215 USA
{nenad,itkis,reyzin,srussell}@cs.bu.edu



**Abstract.** We study the limitations of steganography when the sender is not using any properties of the underlying channel beyond its entropy and the ability to sample from it. On the negative side, we show that the number of samples the sender must obtain from the channel is exponential in the rate of the stegosystem. On the positive side, we present the first secret-key stegosystem that essentially matches this lower bound regardless of the entropy of the underlying channel. Furthermore, for high-entropy channels, we present the first secret-key stegosystem that matches this lower bound *statelessly* (i.e., without requiring synchronized state between sender and receiver).


## 1 Introduction

Steganography's goal is to conceal the presence of a secret message within an innocuous-looking communication. In other words, steganography consists of hiding a secret *hiddentext* message within a public *covertext* to obtain a *stegotext* in such a way that any observer (except, of course, the intended recipient) is unable to distinguish between a covertext *with* a hiddentext and one *without*.

The first rigorous complexity-theoretic formulation of secret-key steganography was provided by Hopper, Langford and von Ahn [HLvA02]. In this formulation, *steganographic secrecy* of a stegosystem is defined as the inability of a polynomial-time adversary to distinguish between observed distributions of unaltered covertexts and stegotexts. (This is in contrast with many previous works, which tended to be information-theoretic in perspective; see, e.g., [Cac98] and other references in [HLvA02,Cac98].)

The model of [HLvA02], which we adopt with slight changes, assumes that the two communicating parties have some underlying channel $\mathcal{C}$ of covertext documents that the adversary expects to see. They also share a secret key (public-key steganography is addressed in [vAH04,BC04]). The sender is allowed to draw documents from $\mathcal{C}$; the game for the sender is to alter $\mathcal{C}$ imperceptibly for the adversary, while transmitting a meaningful hiddentext message to the recipient. Conversely, the game for the (passive) adversary is to distinguish the distribution of transmitted messages from $\mathcal{C}$.

### 1.1 Desirable Characteristics of a Stegosystem

*Black-Box.* In order to obtain a stegosystem of broad applicability, one would like to make as few assumptions as possible about the understanding of the underlying channel. Indeed, as Hopper et al. [HLvA02] point out, the channel (such as human email traffic or images of various scenes) may well be very complex and not easily described. For example, if the parties are using photographs of city scenes as covertexts, it is reasonable to assume that the sender can obtain such photographs, but unreasonable to expect the sender and the recipient to know a polynomial-time algorithm that can construct such photographs from uniformly distributed random strings. In this work, we therefore concentrate on the study of *black-box* steganography. Namely, the sender and the recipient need not know anything about the underlying channel distribution (beyond a lower bound on its min-entropy). The sender's only access to the channel is via an oracle that draws a random sample from the channel distribution. The recipient need not access the channel at all.

*Efficient and Secure.* Stegosystems have several performance characteristics. First, of course, it is desirable that the encoding algorithm of sender and the decoding algorithm of the receiver be efficient. A particularly important characteristic of the efficiency of the sender is the number of samples that the sender is required to draw from $\mathcal{C}$. In fact, in all proposed black-box stegosystems, sender computation is proportional to the number of samples drawn, with actual computation per sample being quite minimal. Because most real-life channels are quite complex, the drawing of the samples is likely to dominate the running time of an actual implementation.

Another important performance measure is the transmission rate of the stegosystem, which is the number of hiddentext bits transmitted per single stegotext document sent (a document is the value returned by a single request to the channel sampling oracle—e.g., a photograph). Transmission rate is tied to reliability, which is the probability of successful decoding of an encoded message (correspondingly, unreliability is one minus reliability). The goal is to construct stegosystems that are reliable and transmit at a high rate (it is, of course, easier to transmit at a high rate if reliability is low and the recipient will not understand much of what is transmitted).

Finally, even a most efficient stegosystem is useless if not secure. Quantitatively, insecurity is defined as the adversary's advantage in distinguishing stegotext from $\mathcal{C}$ (and security as one minus insecurity). Naturally, we are interested in stegosystems with insecurity as close to 0 as possible.

The efficiency and security of a stegosystem, even if it is black-box, may depend on the channel distribution. In particular, we will be interested in the dependence on the channel min-entropy $h$. Ideally, a stegosystem would work well even for low-min-entropy channels.

*Stateless.* It is desirable to construct *stateless* stegosystems, so that the sender and the recipient need not maintain synchronized state in order to communicate long messages. Indeed, the need for synchrony may present a particular problem in steganography in case messages between sender and recipient are dropped or arrive out of order. Unlike in counter-mode symmetric encryption, where the counter value can be sent along with the ciphertext in the clear, here this is not possible: the counter itself would also have to be steganographically encoded to avoid detection, which brings us back to the original problem of steganographically encoding multibit messages.

### 1.2 Our Contributions

We study the optimal efficiency achievable by black-box steganography, and present secret-key stegosystems that are nearly optimal. Specifically, we demonstrate the following results:

- A lower bound, which states that a secure and reliable black-box stegosystem with rate of $w$ bits per document sent requires the encoder to take at least $c2^w$ samples from the channel per $w$ bits sent, for some constant $c$. The value of $c$ depends on security and reliability, and tends to $1/(2e)$ as security and reliability approach 1. This lower bound applies to secret-key as well as public-key stegosystems.
- A stateful black-box secret-key stegosystem STF that transmits $w$ bits per document sent, takes $2^w$ samples per $w$ bits, has unreliability of $2^{-h+w}$ per document, and negligible insecurity, which is independent of the channel. (A very similar construction was independently discovered by Hopper [Hop04, Construction 6.10].)
- A stateless black-box secret-key stegosystem STL that transmits $w$ bits per document sent, takes $2^w$ samples per $w$ bits, has unreliability $2^{-\Theta(2^h)}$, and insecurity negligibly close to $l^2 2^{-h+2w}$ for $lw$ bits sent.

Note that for both stegosystems, the rate vs. number of samples tradeoff is very close to the lower bound—in fact, for channels with sufficient entropy, the optimal rate allowed by the lower bound and the achieved rate differ by $\log_2 2e < 2.5$ bits (and some of that seems due to slack in the bound). Thus, our bound is quite tight, and our stegosystems quite efficient. The proof of the lowerbound involves a surprising application of the huge random objects of [GGN03], specifically of the truthful implementation of a boolean function with interval-sum queries. The lowerbound demonstrates that significant improvements in stegosystem performance must come from assumptions about the channel.

The stateless stegosystem STL can be used whenever the underlying channel distribution has sufficient min-entropy $h$ for the insecurity to be acceptably low. It is extremely simple, requiring just evaluations of a pseudorandom function for encoding and decoding, and very reliable.

If the underlying channel does not have sufficient min-entropy, then the stateful stegosystem STF can be used, because its insecurity is independent of the channel. While it requires shared synchronized state between sender and receiver, the state information is only a counter of the number of documents sent so far. If min-entropy of the channel is so low that the error probability of $2^{-h+w}$ is too high for the application, reliability of this stegosystem can be improved through the use of error-correcting codes over the $2^w$-ary alphabet (applied to the hiddentext before stegoencoding), because failure to decode correctly is independent for each $w$-bit block. Error-correcting codes can increase reliability to be negligibly close to 1 at the expense of reducing the asymptotic rate from $w$ to $w-(h+2)2^{-h+w}$. Finally, of course, the min-entropy of any channel can be improved from $h$ to $nh$ by viewing $n$ consecutive samples as a single draw from the channel; if $h$ is extremely small to begin with, this will be more efficient than using error-correcting codes (this improvement requires both parties to be synchronized modulo $n$, which is not a problem in the stateful case).

This stateful stegosystem STF also admits a few variants. First, the logarithmic amount of shared state can be eliminated at the expense of adding a linear amount of private state to the sender and reducing reliability slightly (as further described in 4.1), thus removing the need for synchronization between the sender and the recipient. Second, under additional assumptions about the channel (e.g., if each document includes time sent, or has a sequence number), STF can be made completely stateless. The remarks of this paragraph and the previous one can be equally applied to [Hop04, Construction 6.10].

### 1.3 Related Work

The bibliography on the subject of steganography is extensive; we do not review it all here, but rather recommend references in [HLvA02].

*Constructions.* In addition to introducing the complexity-theoretic model for steganography, [HLvA02] proposed two constructions of black-box[1] secret-key stegosystems, called Construction 1 and Construction 2.

Construction 1 is stateful and, like our stateful construction STF, boasts negligible insecurity regardless of the channel. However, it can transmit only 1 bit per document, and its reliability is limited by $1/2+1/4(1-2^{-h})$ per document sent, which means that, regardless of the channel, each hiddentext bit has probability at least 1/4 of arriving incorrectly (thus, to achieve high reliability, error-correcting codes with expansion factor of at least $1/(1-H_2(1/4)) \approx 5$ are needed). In contrast, STF has reliability that is exponentially (in the min-entropy) close to 1, and thus works well for any channel with sufficient entropy. Furthermore, it can transmit at rate $w$ for any $w<h$, provided the encoder has sufficient time for the $2^w$ samples required. It can be seen as a generalization of Construction 1.

Construction 2 of [HLvA02] is stateless. Like the security of our stateless construction STL, its security depends on the min-entropy of the underlying channel. While no exact analysis is provided in [HLvA02], the insecurity of Construction

[1] Construction 2, which, strictly speaking, is not presented as a black-box construction in [HLvA02], can be made black-box through the use of extractors (such as universal hash functions) in place of unbiased functions, as shown in [vAH04].

2 seems to be roughly $\sqrt{l}2^{(-h+w)/2}$ (due to the fact that the adversary sees $l$ samples either from $\mathcal{C}$ or from a known distribution with bias roughly $2^{(-h+w)/2}$ caused by a public extractor; see Appendix A), which is higher than the insecurity of STL (unless $l$ and $w$ are so high that $h < 3w + 3\log l$, in which case both constructions are essentially insecure, because insecurity is higher than the inverse of the encoder's running time $l2^w$). Reliability of Construction 2, while not analyzed in [HLvA02], seems close to the reliability of STL. The rate of Construction 2 is lower (if other parameters are kept the same), due to the need for randomized encryption of the hiddentext, which necessarily expands the number of bits sent.

It is important to note that the novelty of STL is not the construction itself, but rather its analysis. Specifically, its stateful variant appeared as Construction 1 in the Extended Abstract of [HLvA02], but the analysis of the Extended Abstract was later found to be flawed by [KMR02]. Thus, the full version of [HLvA02] included a different Construction 1. We simply revive this old construction, make it stateless, generalize it to $w$ bits per document, and, most importantly, provide a new analysis for it.

In addition to the two constructions of [HLvA02] described above, and independently of our work, Hopper in [Hop04] proposed two more constructions: Constructions 6.10 ("`MultiBlock`") and 3.15 ("`NoState`"). As already mentioned, `MultiBlock` is essentially the same as our STF. `NoState` is an interesting variation of Construction 1 of [HLvA02], that addresses the problem of maintaining shared state at the expense of lowering the rate even further.

*Bounds on the Rate and Efficiency.* Hopper in [Hop04, Section 6.2] establishes a bound on the rate vs. efficiency tradeoff. Though quantitatively similar to ours (in fact, tighter by the constant of $2e$), this bound applies only to a restricted class of black-box stegosystems: essentially, stegosystems that encode and decode one block at a time and sample a fixed number of documents per block. The bound presented in this paper applies to any black-box stegosystem, as long as it works for a certain reasonable class of channels, and thus can be seen as a generalization of the bound of [Hop04]. Our proof techniques are quite different than those of [Hop04], and we hope they may be of independent interest. We refer the reader to Section 3.3 for an elaboration. Finally it should be noted that non-black-box stegosystems can be much more efficient—see [HLvA02,vAH04,Le03,LK03].

## 2 Definitions

### 2.1 Steganography

The definitions here are essentially those of [HLvA02]. We modify them in three ways. First, we view the channel as producing documents (symbols in some, possibly very large, alphabet) rather than bits. This simplifies notation and makes min-entropy of the channel more explicit. Second, we consider stegosystem reliability as a parameter rather than a fixed value. Third, we make the length of the adversary's description (and the adversary's dependence on the channel) more explicit in the definition.

*The Channel.* Let $\Sigma$ be an alphabet; we call the elements of $\Sigma$ *documents*. A channel $\mathcal{C}$ is a map that takes a history $\mathcal{H} \in \Sigma^*$ as input and produces a probability distribution $D_{\mathcal{H}} \in \Sigma$. A history $\mathcal{H} = s_1 s_2 ... s_n$ is *legal* if each subsequent symbol is obtainable given the previous ones, i.e., $Pr_{D_{s_1 s_2 \ldots s_{i-1}}}[s_i] > 0$. Min-entropy of a distribution $D$ is defined as $H_\infty(D) = \min_{s \in D}\{-\log_2 \Pr_D[s]\}$. Min-entropy of $\mathcal{C}$ is the $\min_{\mathcal{H}} H_\infty(D_{\mathcal{H}})$, where the minimum is taken over legal histories $\mathcal{H}$.

Our stegosystems will make use of a channel sampling oracle $M$, which, on input $\mathcal{H}$, outputs a symbol $s$ according to $D_{\mathcal{H}}$.

**Definition 1.** *A* black-box secret-key stegosystem *is a pair of probabilistic polynomial time algorithms $S = (SE, SD)$ such that, for a security parameter $\kappa$,*

1. *$SE$ has access to a channel sampling oracle $M$ for a channel $\mathcal{C}$ and takes as input a randomly chosen key $K \in \{0,1\}^\kappa$, a string $m \in \{0,1\}^*$ (called the* hiddentext*), and the channel history $\mathcal{H}$. It returns a string of symbols $s_1 s_2 \ldots s_l \in \Sigma^*$ (called the* stegotext*)*
2. *$SD$ takes as input a key $K \in \{0,1\}^\kappa$, a stegotext $s_1 s_2 \ldots s_l \in \Sigma^*$ and a channel history $\mathcal{H}$, and returns a hiddentext $m \in \{0,1\}^*$.*

*We further assume that the length $l$ of the stegotext output by $SE$ depends only on the length of hiddentext $m$ but not on its contents.*

*Stegosystem Reliability.* The *reliability* of a stegosystem $S$ with security parameter $\kappa$ for a channel $\mathcal{C}$ and messages of length $l$ is defined as

$$\mathbf{Rel}_{S(\kappa),\mathcal{C},l} = \min_{m \in \{0,1\}^l, \mathcal{H}} \{ \Pr_{K \in \{0,1\}^\kappa} [SD(K, SE^M(K, m, \mathcal{H}), \mathcal{H}) = m] \} .$$

Unreliability (as a parallel to insecurity) is defined as $\mathbf{UnRel}_{S(\kappa),\mathcal{C},l} = 1 - \mathbf{Rel}_{S(\kappa),\mathcal{C},l}$.

*The Adversary.* We consider only passive adversaries who mount a chosen hiddentext attack on $S$ (stronger adversarial models for steganography have also been considered, see e.g. [HLvA02,vAH04,BC04]). The goal of such an adversary is to distinguish whether it is seeing encodings of the hiddentext it supplied to the encoder, or simply random draws from the channel. To this end, define an oracle $O(\cdot, \mathcal{H})$ that produces random draws from the channel starting with history $\mathcal{H}$ as follows: on input $m \in \{0,1\}^*$, $O$ computes the length $l$ of the stegotext that $SE^M(K, m)$ would have output, and outputs $s_1 s_2 \ldots s_l$ where each $s_i$ is drawn according to $D_{\mathcal{H} \circ s_1 s_2 \ldots s_{i-1}}$.

**Definition 2.** $W$[2] *is a $(t, d, q, l)$* passive adversary for stegosystem $S$ *if*

[2] The adversary in the context of steganography is sometimes referred to as the "warden." The idea of the adversary as a warden and the use of $W$ to designate it is a consequence of original problem formulation in [Sim83].

1. *$W$ runs in expected time $t$ (including the running time needed by the stegoencoder to answer its queries) and has description of length $d$ (in some canonical language).*
2. *$W$ has access to $\mathcal{C}$ via the sampling oracle $M(\cdot)$.*
3. *$W$ can make an expected number of $q$ queries of combined length $l$ bits to an oracle which is either $SE^M(K,\cdot,\cdot)$ or $O(\cdot,\cdot)$.*
4. *$W$ outputs a bit indicating whether it was interacting with $SE$ or with $O$.*

*Stegosystem Security.* The *advantage* $\mathbf{Adv}^{\mathrm{SS}}$ (here SS stands for "Steganographic Secrecy) of $W$ against $S$ with security parameter $\kappa$ for a channel $\mathcal{C}$ is defined as

$$\mathbf{Adv}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(W) = \left| \Pr_{K \leftarrow \{0,1\}^\kappa}[W^{M,SE^M(K,\cdot,\cdot)} = 1] - \Pr[W^{M,O(\cdot,\cdot)} = 1] \right| .$$

For a given $(t,d,q,l)$, the *insecurity* of a stegosystem $S$ with respect to channel $\mathcal{C}$ is defined as

$$\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(t,d,q,l) = \max_{(t,d,q,l) \text{ adversary } W} \left\{ \mathbf{Adv}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(W) \right\},$$

and security **Sec** as $1 - \mathbf{InSec}$.

Note that the adversary's algorithm can depend on the channel $\mathcal{C}$, subject to the restriction on the algorithm's total length $d$. In other words, the adversary can possess some description of the channel in addition to the black-box access provided by the channel oracle. This is a meaningful strengthening of the adversary: indeed, it seems imprudent to assume that the adversary's knowledge of the channel is limited to whatever is obtainable by black-box queries (for instance, the adversary has some idea of a reasonable email message or photograph should look like). It does not contradict our focus on black-box steganography: it is prudent for the honest parties to avoid relying on particular properties of the channel, while it is perfectly sensible for the adversary, in trying to break the stegosystem, to take advantage of whatever information about the channel is available.

### 2.2 Pseudorandom Functions

We use pseudorandom functions [GGM86] as a tool. Because the adversary in our setting has access to the channel, any cryptographic tool used must be secure even given the information provided by the channel. Thus, our underlying assumption is the existence of pseudorandom functions that are secure given the channel oracle, which is equivalent [HILL99] to the existence of one-way functions that are secure given the channel oracle. Thus is the minimal assumption needed for steganography [HLvA02].

Let $\mathcal{F} = \{F_{\mathrm{seed}}\}_{\mathrm{seed} \in \{0,1\}^*}$ be a family of functions, all with the same domain and range. For a probabilistic adversary $A$, and channel $\mathcal{C}$ with sampling oracle $M$, the *PRF-advantage of $A$ over $\mathcal{F}$* is defined as

$$\mathbf{Adv}^{\mathrm{PRF}}_{\mathcal{F}(n),\mathcal{C}}(A) = \left| \Pr_{\mathrm{seed} \leftarrow \{0,1\}^n}[A^{M,F_{\mathrm{seed}}(\cdot)} = 1] - \Pr_g[A^{M,g(\cdot)} = 1] \right| ,$$

where $g$ is a random function with the same domain and range. For a given $(t, d, q)$, the *insecurity* of a pseudorandom function family $\mathcal{F}$ with respect to channel $\mathcal{C}$ is defined as

$$\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n),\mathcal{C}}(t,d,q,l) = \max_{(t,d,q,l) \text{ adversary } A} \{\mathbf{Adv}^{\mathrm{SS}}_{\mathcal{F}(n),\mathcal{C}}(A)\},$$

where the maximum is taken over all adversaries that run in expected time $t$, whose description size is at most $d$, and that make an expected number of $q$ queries to their oracles.

## 3 The Lower Bound

Recall that we define the rate of a stegosystem as the *average number of hiddentext bits per document sent* (this should not be confused with the average number of hiddentext bits per *bit* sent; note also that this is the sender's rate, not the rate of information actually decoded by the recipient, which is lower due to unreliability). We set out to prove that a reliable stegosystem with black-box access to the channel with rate $w$, must make roughly $l2^w$ queries to the channel to send a message of length $lw$. Intuitively, this should be true because each document carries $w$ bits of information on average, but since the encoder knows nothing about the channel, it must keep on sampling until it gets the encoding of those $w$ bits, which amounts to $2^w$ samples on average.

In particular, it suffices for the purposes of this lower bound to consider a restricted class of channels: the distribution of the sample depends only on the length of the history (not on its contents). We will write $D_1, D_2, ..., D_i, ...$, instead of $D_{\mathcal{H}}$, where $i$ is the length of the history $\mathcal{H}$. Furthermore, it will suffice for us to consider only distributions $D_i$ that are uniform on a subset of $\Sigma$. We will identify the distribution with the subset (as is often done for uniform distributions).

Let $|D_i| = H = 2^h$ and $|\Sigma| = S$. Because the encoder receives the min-entropy $h$ of the channel as input, if $H = S$, then encoder knows the channel completely (it's simply uniform on $\Sigma$), and our lower bounds do not hold, because no sampling from the channel is necessary. Thus, we require that $h$ be smaller than $\log_2 S$. Let $R = 1/(1 - H/S)$.

Our proof proceeds in two parts. First, we consider a stegoencoder $SE$ that does not output anything that it did not receive as a response from the channel-sampling oracle. To be reliable, such an encoder has to make many queries, as shown in Lemma 1. Second, we show that to be secure, a black-box $SE$ cannot output anything it did not receive from the channel-sampling oracle.

The second half of the proof is somewhat complicated by the fact that we want to assume security only against bounded adversaries: namely, ones whose description size and running time are polynomial in the description size and running time of the encoder (in particular, polynomial in $\log S$ rather than $S$). This requires us to come up with pseudorandom subsets $D_i$ of $\Sigma$ that have concise descriptions and high min-entropy, and whose membership is impossible for the stegoencoder to predict. In order to do that, we utilize techniques from the truthful implementation of a boolean function with interval-sum queries of [GGN03] (truthfulness is important because min-entropy has to be high unconditionally).

### 3.1 Lower Bound When Only Query Results Are Output

We consider the following channel: if $D_1, D_2, \ldots$ are subsets of $\Sigma$, we write $\boldsymbol{D} = D_1 \times D_2 \times \ldots$ to denote the channel that, on history length $i$, outputs an uniformly random element of $D_i$; if $|D_1| = |D_2| = \ldots = 2^h$ then we say that $\boldsymbol{D}$ is a *flat $h$-channel*. Normally, one would think of the channel sampling oracle for $\boldsymbol{D}$ as making a fresh random choice from $D_i$ when queried on history length $i$. Instead, we will think of the oracle as having made all its choices in advance. Imagine that the oracle already took "enough" samples:

$$\begin{array}{l} s_{1,1},\ s_{1,2}, \ldots, s_{1,j}, \ldots \text{ from } D_1, \\ s_{2,1},\ s_{2,2}, \ldots, s_{2,j}, \ldots \text{ from } D_2, \\ \qquad\qquad \ldots, \\ s_{i,1},\ s_{i,2}, \ldots, s_{i,j}, \ldots \text{ from } D_i \\ \qquad\qquad \ldots . \end{array}$$

We will denote the string containing all these samples by $\mathcal{S}$, and refer to it as a *draw-sequence* from the channel. We will give our stegoencoder access to an oracle (also denoted by $\mathcal{S}$) that, each time it's queried with $i$, returns the next symbol from the sequence $s_{i,1}, s_{i,2}, \ldots, s_{i,j}, \ldots$. Choosing $\mathcal{S} \in \Sigma^{**}$ at random and giving the stegoencoder access to it is equivalent to giving the encoder access to the usual channel-sampling oracle $M$ for our channel $\boldsymbol{D}$.

Assume $SE^{\mathcal{S}}(K, m, \mathcal{H}) = t = t_1 t_2 \ldots t_l$, where $t_i \in \Sigma$. Note that $t_i$ is an element of the sequence $s_{i,1}, s_{i,2}, \ldots, s_{i,j}, \ldots$. If $t_i$ is the $j$-th element of this sequence, then it took $j$ queries to produce it. We will denote by *weight of $t$ with respect to $\mathcal{S}$*, the number of queries it took to produce $t$: $W(t, \mathcal{S}) = \sum_{i=1}^{k} \min\{j \mid s_{i,j} = y_i\}$. In the next lemma, we prove (by looking at the *decoder*) that for any $\mathcal{S}$, most messages have high weight.

**Lemma 1.** *Let $F : \Sigma^* \to \{0,1\}^*$ be an arbitrary (possibly unbounded) deterministic stegodecoder that takes a sequence $t \in \Sigma^l$ and outputs a message $m$ of length $lw$ bits.*

*Then the probability that a random $lw$-bit message has an encoding of weight significantly less than $(1/e)l2^w$, is small. More precisely, for any $\mathcal{S} \in \Sigma^{**}$ and any $N \in \mathbb{N}$:*

$$\Pr_{m\in\{0,1\}^{lw}}[(\exists t \in \Sigma^l)(F(t) = m \ \wedge\ W(t, \mathcal{S}) \le N)] \le \frac{\binom{N}{l}}{2^{lw}} < \left(\frac{Ne}{l2^w}\right)^l .$$

*Proof.* Simple combinatorics show that the number of different sequences $t$ that have weight less than $N$ (and hence the number of messages that have encodings of weight less than $N$) is at most $\binom{N}{l}$: indeed, it is simply the number of positive integer solutions to $x_1 + \ldots + x_l \le N$, which is the number of ways to put $l$ bars among $N-l$ stars (the number of stars to the right of the $i$-th bar corresponds to $x_i - 1$), or, equivalently, the number of ways choose $l$ positions out of $N$. The total number of messages is $2^{lw}$. The last inequality follows from $\binom{N}{l} < \left(\frac{Ne}{l}\right)^l$. □

Observe that taking the probability over a random $lw$-bit message, as we do above, is meaningful. Indeed, if the distribution of messages encoded is not uniform, then compression could reduce their size and thus improve the efficiency of the stegosystem, rendering our bound pointless. Our lower bound applies when the designer of the stegosystem assumes that the messages are distributed uniformly. (For any other distribution, data compression should be applied before stegoencoding.)

### 3.2 Secure Stegosystems Almost Always Output Query Answers

The next step is to prove that the encoder of a secure black-box stegosystem must output only what it gets from the oracle, with high probability. Assume $\boldsymbol{D}$ is a flat $h$-channel chosen uniformly at random. Then it is easy to demonstrate that, if the encoder outputs in position $i$ a symbol $s_i \in \Sigma$ that it did not receive as a response to a query to $D_i$, the chances that $s_i$ is in the support of $D_i$ are $H/S$. It can then be shown that, if the stegoencoder has insecurity $\epsilon$, then it cannot output something it did not receive as response to a query with probability higher than $\epsilon/(1-H/S)$.

The problem with the above argument is the following: it assumes that the adversary can test whether $s_i$ the support of $D_i$. This is not possible if we assume $D_i$ is completely random and the adversary's description is small compared to $S = |\Sigma|$. However, it does serve as a useful warm-up, and leads to the following theorem when combined with the results of the previous section.

**Theorem 1.** *Let $(SE, SD)$ be a black-box stegosystem with insecurity $\epsilon$ against an adversary who has an oracle for testing membership in the support of $\mathcal{C}$, unreliability $\rho$ and rate $w$ for an alphabet $\Sigma$ of size $S$. Then there exists a channel with min-entropy $h = \log_2 H$ such that the probability that the encoder makes at most $N$ queries to send a random message of length $lw$, is upper bounded by*

$$\left(\frac{Ne}{l2^w}\right)^l + \rho + \epsilon R\,,$$

*and the expected number of queries per stegotext symbol is therefore at least*

$$\frac{2^w}{e}\left(\frac{1}{2} - \rho - \epsilon R\right)\,,$$

*where $R = 1/(1-H/S)$.*

*Proof.* See the full version [DIRR04]. □

### 3.3 Lower Bound for Computationally Bounded Parties

We now want to establish the same lower bound without making such a strong assumption about the security of the stegosystem. Namely, we do not want to assume that the insecurity $\epsilon$ is low unless the adversary's description size and running time are small ("small," when made rigorous, will mean some fixed polynomials in the description size and running time, respectively, of the stegoencoder, and a security parameter for a function that is pseudorandom against the stegoencoder). Recall that our definitions allow the adversary to depend on the channel; thus, our goal is to construct channels that have short descriptions for the adversary but look like random flat $h$-channels to the black-box stegoencoder. In other words, we wish to replace a random flat $h$-channel with a pseudorandom one.

We note that the channel is pseudorandom only in the sense that it has a short description, so as to allow the adversary to be computationally bounded. The min-entropy guarantee, however, can not be replaced with a "pseudo-guarantee": else the encoder is being lied to, and our lower bound is no longer meaningful. Thus, a simpleminded approach, such as using a pseudorandom predicate with bias $H/S$ applied to each symbol and history length to determine whether the symbol is in the support of the channel, will not work here: because $S$ is constant, eventually (for some history length) the channel will have lower than guaranteed min-entropy (moreover, we do not wish to assume that $S$ is large in order to demonstrate that this is unlikely to happen; our lower bound should work for any alphabet). Rather, we need the pseudorandom implementation of the channel to be truthful[3] in the sense of [GGN03], and so rely on the techniques developed therein.

The result is the following theorem.

**Theorem 2.** *There exist polynomials $p, q$ and constants $c_1, c_2$ with the following property. Let $S(\kappa)$ be a black-box stegosystem with description size $\delta$, insecurity $\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(t, d, q, l)$, unreliability $\rho$, rate $w$ and running time $\tau$ for an alphabet $\Sigma$ of size $S$. Assume there exists a pseudorandom function family $\mathcal{F}(n)$ with insecurity $\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(t, d, q)$. Then there exists a channel $\mathcal{C}$ with min-entropy $h = \log_2 H$ such that the probability that the encoder makes at most $N$ queries to send a random message of length $lw$, is upper bounded by*

$$\left(\frac{Ne}{l2^w}\right)^l + \rho + R\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(q(\tau), n + c_1, 1, lw) + (R+1)\left(\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(p(\tau), \delta + c, p(\tau)) + 2^{-n}\right),$$

*and the expected number of queries per stegotext symbol is therefore at least*

$$\frac{2^w}{e}\left(\frac{1}{2} - \rho - R\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(q(\tau), n + c_1, 1, lw)\right) -$$

[3] In this case, truthfulness implies that for each history length, the support of the channel has exactly $H$ elements.

$$\frac{2^w}{e}(R+1)\left(\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(p(\tau),\delta+c,p(\tau))+2^{-n}t\right),$$

*where* $R = 1/(1-H/S)$.

*Proof.* See the full version [DIRR04]. □

*Discussion.* The proof of Theorem 2 relies fundamentally on Theorem 1. In other words, to prove a lower bound in the computationally bounded setting, we use the corresponding lower bound in the information-theoretic setting. To do so, we replace an object of an exponentially large size (the channel) with one that can be succinctly described. This replacement substitutes *some* information-theoretic properties with their computational counterparts. However, for a lower bound to remain "honest" (i.e., not restricted to uninteresting channels), some global properties must remain information-theoretic. This is where the truthfulness of huge random objects of [GGN03] comes to the rescue. We hope that other interesting impossibility results can be proved in a similar fashion, by adapting an information-theoretic result using the paradigm of [GGN03]. We think truthfulness of the objects will be important in such adaptations for the same reason it was important here.

Note that the gap in the capabilities of the adversary and encoder/decoder is different in the two settings: in the information-theoretic case the adversary is given unrestricted computational power, while in the computationally bounded case it is assumed to run in polynomial time, but is given the secret channel seed. However, in the information-theoretic case we may remove the gap altogether, by providing both the adversary and the encoder/decoder with a channel membership oracle, and still obtain a lower bound analogous[4] to that of Theorem 2. We see no such opportunity to remove the gap in the computationally bounded case (e.g., equipping the encoder/decoder with the channel seed seems to break our proof). Removing this asymmetry in the computationally bounded case seems challenging and worth pursuing.

## 4 The Stateful Construction STF

The construction STF relies on a pseudorandom function family $\mathcal{F}$. In addition to the security parameter $\kappa$ (the length of the PRF key $K$), it depends on the rate parameter $w$. Because it is stateful, both encoder and decoder take a counter *ctr* as input.

Our encoder is similar to the rejection-sampler-based encoder of [HLvA02] generalized to $w$ bits: it simply samples elements from the channel until the pseudorandom function evaluated on the element produces the $w$-bit symbol being encoded. The crucial difference of our construction is the following: to

[4] A lower bound on the number of samples per document sent, becomes trivially zero if the encoder is given as much time as it pleases, in addition to the membership oracle of the flat channel. Yet it should not be difficult to prove that it must then run for $O(2^w)$ steps per document sent.

avoid introducing bias into the channel, if the same element is sampled twice, the encoder simply flips a random coin to decide whether to output that element with probability $2^{-w}$. Hopper in [Hop04, Construction 6.10] independently proposes a similar construction, except instead of flipping a fresh random coin, the encoder evaluates the pseudorandom function on a new counter value (there is a separate counter associated to each sampled document, indicating how many times the document has been sampled), thus conserving randomness.

Observe that, assuming $\mathcal{F}$ is truly random rather than pseudorandom, each sample from the channel has probability $2^{-w}$ of being output, independent of anything else, because each time fresh randomness is being used. Of course, this introduces unreliability, which is related to the probability of drawing the same element from $D_{\mathcal{H}}$ twice.

**Procedure** STF.$SE(K, w, m, \mathcal{H}, ctr)$:
    Let $m = m_1 \dots m_l$, where $|m_i| = w$
    `for` $i \leftarrow 1$ `to` $l$:
        $j \leftarrow 0$; $f \leftarrow 0$; $ctr \leftarrow ctr + 1$
        `repeat` :
            $j \leftarrow j + 1$
            $s_{i,j} \leftarrow M(\mathcal{H})$
            `if` $\exists j' < j$ s.t. $s_{i,j} = s_{i,j'}$
                `let` $c \in_R \{0,1\}^w$
                `if` $c = m_i$ `then` $f \leftarrow 1$
            `else if` $F_K(ctr, s_{i,j}) = m_i$
                `then` $f \leftarrow 1$
        `until` $f = 1$
        $s_i \leftarrow s_{i,j}$; $\mathcal{H} \leftarrow \mathcal{H} || s_i$
    `output` $s = s_1 s_2 \dots s_l$

**Procedure** STF.$SD(K, w, s, ctr)$:
    Let $s = s_1 \dots s_l$, where $s_i \in \Sigma$
    `for` $i = 1$ `to` $l$
        $ctr \leftarrow ctr + 1$
        $m_i \leftarrow F_K(ctr, s_i)$
    `output` $m = m_1 m_2 \cdots m_l$

**Theorem 3.** *The stegosystem* STF *has insecurity* $\mathbf{InSec}^{\mathrm{SS}}_{\mathrm{STF}(\kappa,w)}(t, d, l, lw) = \mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(t + O(1), d + O(1), l2^w)$. *For each* $i$, *the probability that* $s_i$ *is decoded incorrectly is* $2^{-h+w} + \mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(2^w, O(1), 2^w)$, *and unreliability is at most* $l(2^{-h+w} + \mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(2^w, O(1), 2^w))$.

*Proof.* Insecurity bound is apparent from the fact that if $\mathcal{F}$ were truly random, then the system would be perfectly secure, because its output is distributed identically to $\mathcal{C}$ (simply because the encoder samples from the channel, and independently at random decides which sample to output, because the random function is never applied more than once to the same input). Hence, any adversary for the stegosystem would distinguish $\mathcal{F}$ from random.

The reliability bound per symbol can be demonstrated as follows. Assuming $\mathcal{F}$ is random, the probability that $s_i = s_{i,j}$ is $(1 - 2^{-w})^{j-1} 2^{-w}$. If that happens, the probability that $\exists j' < j$ such that $s_{i,j} = s_{i,j'}$ is at most $(j-1)2^{-h}$. Summing up and using standard formulas for geometric series, we get

$$\sum_{j=1}^{\infty}(j-1)2^{-h}\left(1-2^{-w}\right)^{j-1}2^{-w} =$$

$$= 2^{-h-w}\sum_{j=1}^{\infty}\left(\left(1-2^{-w}\right)^{j}\left(\sum_{k=0}^{\infty}(1-2^{-w})^{k}\right)\right) < 2^{w-h}.$$

□

Note that errors are independent for each symbol, and hence error-correcting codes over alphabet of size $2^w$ can be used to increase reliability: one simply encodes $m$ before feeding it to $SE$. Observe that, for a truly random $\mathcal{F}$, if an error occurs in position $i$, the symbol decoded is uniformly distributed among all elements of $\{0,1\}^w - \{m_i\}$. Therefore, the stegosystem creates a $2^w$-ary symmetric channel with error probability $2^{w-h}(1-2^{-w}) = 2^{-h}(2^w-1)$ (this comes from more careful summation in the above proof). Its capacity is $w - H[1-2^{-h}(2^w-1), 2^{-h}, 2^{-h}, \ldots, 2^{-h}]$ (where $H$ is Shannon entropy of a distribution) [McE02, p. 58]. This is equal to $w+(2^w-1)2^{-h}\log 2^{-h}+(1-2^{-h}(2^w-1))\log(1-2^{-h}(2^w-1))$. Assuming error probability $2^{-h}(2^w-1) \leq 1/2$ and using $\log(1-x) \geq -2x$ for $0 \leq x \leq 1/2$, we get that the capacity of the channel created by the encoder is at least $w + 2^{-h}(2^w-1)(-h-2) \geq w-(h+2)2^{-h+w}$. Thus, as $l$ grows, we can achieve rates close to $w-(h+2)2^{-h+w}$ with near perfect security and reliability (independent of $h$).

### 4.1 Stateless Variants of STF

Our stegosystem STF is stateful because we need $F$ to take $ctr$ as input, to make sure we never apply the pseudorandom function more than once to the same input. This will happen automatically, without the need for $ctr$, if the channel $\mathcal{C}$ has the following property: for any histories $\mathcal{H}$ and $\mathcal{H}'$ such that $\mathcal{H}$ is the prefix of $\mathcal{H}'$, the supports of $D_{\mathcal{H}}$ and $D_{\mathcal{H}'}$ do not intersect. For instance, when documents have monotonically increasing sequence numbers or timestamps, no shared state is needed.

To remove the need for shared state for all channels, we can do the following. We remove $ctr$ as an input to $F$, and instead provide STF.$SE$ with the set $Q$ of all values received so far as answers from $M$. We replace the line "`if` $\exists j' < j$ s.t. $s_{i,j} = s_{i,j'}$" with "`if` $s_{i,j} \in Q$" and add the line "$Q \leftarrow Q \cup \{s_{i,j}\}$" before the end of the inner loop. Now shared state is no longer needed for security, because we again get fresh coins on each draw from the channel, even if it collides with a draw made for a previous hiddentext symbol. However, reliability suffers, because the larger $l$ is, the more likely a collision will happen. A careful analysis, omitted here, shows that unreliability is $l^2 2^{-h+w}$ (plus the insecurity of the PRF).

Unfortunately, this variant requires the encoder to store the set $Q$ of all the symbols ever sampled from $\mathcal{C}$. Thus, while it removes shared state, it requires a lot of private state. This storage can be reduced somewhat by use of Bloom filters [Blo70] at the expense of introducing potential false collisions and thus further decreasing reliability. An analysis utilizing the bounds of [BM02] (omitted here) shows that using a Bloom filter with $(h - w - \log l)/\ln 2$ bits per entry will increase unreliability by only a factor of 2, while potentially reducing storage significantly (because the symbols of $\Sigma$ require at least $h$ bits to store, and possibly more if the $D_{\mathcal{H}}$ is sparse).

## 5 The Stateless Construction STL

The stateless construction STL is simply STF without the counter and collision detection (and is a generalization to rate $w$ of the construction that appeared in the extended abstract of [HLvA02]). Again, we emphasize that the novelty is not in the construction but in the analysis. The construction requires a reliability parameter $k$, to make sure that expected running time of the encoder does not become infinite due a low-probability event of infinite running time.

**Procedure** STL.$SE(K, w, k, m, \mathcal{H})$:
```
Let m = m_1 ... m_l, where |m_i| = w
for i ← 1 to l:
    j ← 0
    repeat :
        j ← j + 1
        s_{i,j} ← M(H)
    until F_K(s_{i,j}) = m_i or j = k
    s_i ← s_{i,j}; H ← H||s_i
output s = s_1 s_2 ... s_l
```

**Procedure** STL.$SD(K, w, s)$:
```
Let s = s_1 ... s_l, where s_i ∈ Σ
for i = 1 to l
    m_i ← F_K(s_i)
output m = m_1 m_2 ··· m_l
```

**Theorem 4.** *The stegosystem* STL *has insecurity*

$$\mathbf{InSec}^{\mathrm{SS}}_{\mathrm{STL}(\kappa,w,k),\mathcal{C}}(t, d, l, lw) \in O(2^{-h+2w} l^2 + l e^{-k/2^w}) + \mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(t + O(1), d + O(1), l2^w)\,.$$

*More precisely,*

$$\begin{aligned}\mathbf{InSec}^{\mathrm{SS}}_{\mathrm{STL}(\kappa,w,k),\mathcal{C}}(t, d, l, lw) < 2^{-h}\left(l(l+1)2^{2w} - l(l+3)2^w + 2l\right)\\ +2l\left(1 - \frac{1}{2^w}\right)^k\\ +\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(t + 1, d + O(1), l2^w).\end{aligned}$$

*Proof.* The proof of Theorem 4 consists of a hybrid argument. The first step in the hybrid argument is replace the stegoencoder $SE$ with $SE_1$, which is the same as $SE$ except that it uses a truly random $G$ instead of pseudorandom $F$, which accounts for the term $\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(t + O(1), d + O(1), l2^w)$. Then, rather than consider directly the statistical difference between $\mathcal{C}$ and the output of

$SE_1$ on an $lw$-bit message, we bound it via a series of steps involving related stegoencoders (these are not encoders in the sense defined in Section 2, as they do not have corresponding decoders; they are simply related procedures that help in the proof).

We now describe these encoders $SE_2$, $SE_3$, and $SE_4$. $SE_2$ is the same as $SE_1$, except that it maintains a set $Q$ of all answers received from $M$ so far. After receiving an answer $s_{i,j} \leftarrow M(\mathcal{H})$, it checks if $s_{i,j} \in Q$; if so, it aborts and outputs "Fail"; else, it adds $s_{i,j}$ to $Q$. It also aborts and outputs "Fail" if $j$ ever reaches $k$ during an execution of the inner loop. $SE_3$ is the same as $SE_2$, except that instead of thinking of random function $G$ as being fixed before hand, it creates $G$ "on the fly" by repeatedly flipping coins to decide the $w$-bit value assigned to $s_{i,j}$. Since, like $SE_2$, it aborts whenever a collision between strings of covertexts occurs, the function will remain consistent. Finally, $SE_4$ is the same as $SE_3$, except that it never aborts with failure.

In a sequence of lemmas, we bound the statistical difference between the outputs of $SE_1$ and $SE_2$; show that it is the same as the statistical difference between the outputs of $SE_3$ and $SE_4$; and show that the outputs of $SE_2$ and $SE_3$ are distributed identically. Finally, observe that $SE_4$ does nothing more than sample from the channel and then randomly and obliviously to the sample keep or discard it. Hence, its output is distributed identically to the channel. The details of the proof are contained in the full version [DIRR04]. □

**Theorem 5.** *The stegosystem* STL *has unreliability*

$$\mathbf{UnRel}^{\mathrm{SS}}_{\mathrm{STL}(\kappa,w,k),\mathcal{C},l} \leq l\left(2^w \exp\left[-2^{h-2w-1}\right] + \exp\left[-2^{-w-1}k\right]\right) + \mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(\kappa)}(t, d, l2^w)\,,$$

*where $t$ and $d$ are the expected running time and description size, respectively, of the stegoencoder and the stegodecoder combined.*

*Proof.* As usual, we consider unreliability if the encoder is using a truly random $G$; then, for a pseudorandom $F$, the encoder and decoder will act as a distinguisher for $F$ (because whether something was encoded correctly can be easily tested by the decoder), which accounts for the $\mathbf{InSec}^{PRF}$ term.

Now, fix channel history $\mathcal{H}$ and $w$-bit message $m$, and consider the probability that $G(D_\mathcal{H})$ is so skewed that the weight of $G^{-1}(m)$ in $D_\mathcal{H}$ is less $c2^{-w}$ for some constant $c < 1$ (note that the expected weight is $2^{-w}$). Let $\Sigma = \{s_1 \ldots s_n\}$ be the alphabet, and let $\Pr_{D_\mathcal{H}}[s_i] = p_i$. Define random variable $X_i$ as $X_i = 0$ if $G(s_i) = m$ and $X_i = p_i$ otherwise. Then the weight of $G^{-1}(m)$ equals $1 - \sum_{i=1}^n X_i$. Note that the expected value of $\sum_{i=1}^n X_i = 1 - 2^{-w}$. Using Hoeffding's inequality (Theorem 2 of [Hoe63]), we obtain

$$\begin{aligned}\Pr[1 - \sum_{i=1}^n X_i \leq cR] &\leq \exp\left[-2(1-c)^2 2^{-2w} / \sum_{i=1}^n p_i^2\right] \\ &\leq \exp\left[-2(1-c)^2 2^{-2w} / 2^{-h} / \sum_{i=1}^n p_i\right]\end{aligned}$$

$$= \exp\left[-2(1-c)^2 2^{h-2w}\right] ,$$

where the second to last step follows from $p_i \le 2^{-h}$ and the last step follows from $\sum_{i=1}^{n} p_i = 1$. If we now set $c = 1/2$ and take the union bound over all message $m \in \{0,1\}^w$, we get $2^w \exp\left[-2^{h-2w-1}\right]$.

Assuming $G(D_{\mathcal{H}})$ is not so skewed, the probability of failure is

$$(1 - c2^{-w})^k \le \exp\left[-c2^{-w}k\right] .$$

The result follows from the union bound over $l$. □

## Acknowledgements

We are grateful to Nick Hopper for clarifying related work.

The authors were supported in part by the National Science Foundation under Grant No. CCR-0311485. Scott Russell's work was also facilitated in part by a National Physical Science Consortium Fellowship and by stipend support from the National Security Agency.

## References

[BC04] Michael Backes and Christian Cachin. Public-key steganography with active attacks. Technical Report 2003/231, Cryptology e-print archive, `http://eprint.iacr.org`, 2004.

[Blo70] B. Bloom. Space/time tradeoffs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[BM02] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. In *Proceedings of the Fortieth Annual Allerton Conference on Communication, Control and Computing*, 2002.

[Cac98] C. Cachin. An information-theoretic model for steganography. In *Second Internation Workshop on Information Hiding*, volume 1525 of *Lecture Notes in Computer Science*, pages 306–316, 1998.

[DIRR04] Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. Upper and lower bounds on black-box steganography. Technical Report 2004/246, Cryptology e-print archive, `http://eprint.iacr.org`, 2004.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GGN03] Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. In *44th Annual Symposium on Foundations of Computer Science*, pages 68–79, Cambridge, Massachusetts, October 2003.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HLvA02] N. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. Technical Report 2002/137, Cryptology e-print archive, `http://eprint.iacr.org`, 2002. Preliminary version in Crypto 2002.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[Hop04] Nicholas J. Hopper. *Toward a Theory of Steganography*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, July 2004. Available as Technical Report CMU-CS-04-157.

[KMR02] Lea Kissner, Tal Malkin, and Omer Reingold. Private communication to N. Hopper, J. Langford, L. von Ahn, 2002.

[Le03] Tri Van Le. Efficient provably secure public key steganography. Technical Report 2003/156, Cryptology e-print archive, `http://eprint.iacr.org`, 2003.

[LK03] Tri Van Le and Kaoru Kurosawa. Efficient public key steganography secure against adaptively chosen stegotext attacks. Technical Report 2003/244, Cryptology e-print archive, `http://eprint.iacr.org`, 2003.

[McE02] Robert J. McEliece. *The Theory of Information and Coding*. Camridge University Press, second edition, 2002.

[Rey04] Leonid Reyzin. A Note On the Statistical Difference of Small Direct Products. Technical Report BUCS-TR-2004-032, CS Department, Boston University, September 21 2004. Available from `http://www.cs.bu.edu/techreports/`.

[Sim83] G. J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, *Advances in Cryptology: Proceedings of Crypto 83*, pages 51–67. Plenum Press, New York and London, 1984, 22–24 August 1983.

[vAH04] Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

## A On Using Public $\varepsilon$-Biased Functions

Many stegosystems [HLvA02,vAH04,BC04] (particularly public-key ones) use the following approach: they encrypt the plaintext using encryption that is indistinguishable from random, and then use rejection sampling with a public function $f : \Sigma \to \{0,1\}^w$ to stegoencode the plaintext.

For security, $f$ should have small bias on $D_{\mathcal{H}}$: i.e., for every $c \in \{0,1\}^w$, $\Pr_{s \in D_{\mathcal{H}}}[s \in f^{-1}(c)]$ should be close to $2^{-w}$. It is commonly suggested that a universal hash function with a published seed (e.g., as part of the public key) be used for $f$.

Assume the stegosystem has to work with a memoryless channel $\mathcal{C}$, i.e., one for which the distribution $D$ is the same regardless of history. Let $E$ be the distribution induced on $\Sigma$ by the following process: choose a random $c \in \{0,1\}^w$ and then keep choosing $s \in D$ until $f(s) = c$. Note that the statistical difference between $D$ and $E$ is exactly the bias $\varepsilon$ of $f$. We are interested in the statistical difference between $D^l$ and $E^l$.

For a universal hash function $f$ that maps a distribution of min-entropy $h$ to $\{0,1\}^w$, the bias is roughly $\varepsilon = 2^{(-h+w)/2}$. As shown in [Rey04], if $l < 1/\varepsilon$ (which is reasonable to assume here), statistical difference between $D^l$ and $E^l$ is roughly at least $\sqrt{l}\varepsilon$.

Hence, the approach based on public hash functions results in statistical insecurity of about $\sqrt{l}2^{(-h+w)/2}$.