Upper and Lower Bounds on Black-Box Steganography*

Nenad Dedić Gene Itkis Leonid Reyzin Scott Russell

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

July 9, 2006

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 an unauthorized observer 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 [10]. 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., [4] and other references in [10, 4].)

1.1 Model

In steganography, the very presence of a message must be hidden from the adversary, who must be given no reason for suspecting that anything is unusual. This is the main difference from encryption, which does not prevent the adversary from suspecting that a secret message is being sent, but only from decoding the message. To formalize "unusual," some notion of usual communication must exist.

*Preliminary version appears in TCC 2005 [5].

We adopt the model of [10] with minor changes. In it, *sender* sends data to *receiver*. The usual (nonsteganographic) communication comes from the *channel*, which is a distribution of possible *documents* sent from sender to receiver based on past communication. The channel models the sender's decision process about what to say next in ordinary communication; thus, the sender is given access to the channel via a *sampling oracle* that takes the past communication as input and returns the next document from the appropriate probability distribution. Sender and receiver share a secret key (public-key steganography is addressed in [17, 1]).

The adversary is assumed to also have some information about the usual communication, and thus about the channel. It listens to the communication and tries to distinguish the case when the sender and receiver are just carrying on the usual conversation (equivalently, sender is honestly sampling from the oracle) from the case when the sender is transmitting a hiddentext message $m \in \{0,1\}^*$ (the message may even be chosen by the adversary). A stegosystem is secure if the adversary's suspicion is not aroused—i.e., if the two cases cannot be distinguished.

1.2 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. As Hopper et al. [10] point out, the channel may 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 receiver to know a polynomial-time algorithm that can construct such photographs from uniformly distributed random strings. We therefore concentrate on *black-box* steganography, in which the knowledge about the channel is limited to the sender's ability to query the sampling oracle and a bound on the channel's min-entropy available to sender and receiver. In particular, the receiver is not assumed to be able to sample from the channel. The adversary, of course, may know more about the channel.

Efficient (in terms of running time, number of samples, rate, reliability). The running times of sender's and receiver's algorithms should be minimized. Affairs are slightly complicated by the sender's algorithm, which involves two kinds of fundamentally different operations: *computation*, and *channel sampling*. Because obtaining a channel sample could conceivably be of much higher cost than performing a computation step, the two should be separately accounted for.

Transmission rate of a stegosystem is the number of hiddentext bits transmitted per single stegotext document sent. Transmission rate is tied to *reliability*, which is the probability of successful decoding of an encoded message (and *unreliability*, which is one minus reliability). The goal is to construct stegosystems that are reliable and transmit at a high rate (it is easier to transmit at a high rate if reliability is low and so the receiver will not understand much of what is transmitted).

Even if a stegosystem is black-box, its efficiency may depend on the channel distribution. 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.

Secure. *Insecurity* is defined as the adversary's advantage in distinguishing stegotext from regular channel communication (and *security* as one minus insecurity). Note that security, like efficiency, may depend on the channel min-entropy. We are interested in stegosystems with insecurity as close to 0 as possible, ideally even for low-min-entropy channels.

Stateless. It is desirable to construct *stateless* stegosystems, so that the sender and the receiver 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 receiver 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.

1.3 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 [11, 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 [7], 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 receiver. 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 [11, Construction 6.10].

1.4 Related Work

The bibliography on the subject of steganography is extensive; we do not review it all here, but rather recommend references in [10].

Constructions. In addition to introducing the complexity-theoretic model for steganography, [10] 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 [10] 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 [10], the insecurity of Construction 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 [10], 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 [10], but the analysis of the Extended Abstract was later found to be flawed by [12]. Thus, the full version of [10] 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.

[1] Construction 2, which, strictly speaking, is not presented as a black-box construction in [10], can be made black-box through the use of extractors (such as universal hash functions) in place of unbiased functions, as shown in [17].

In addition to the two constructions of [10] described above, and independently of our work, Hopper in [11] 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 [10], 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 [11, 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 [11]. Our proof techniques are quite different than those of [11], and we hope they may be of independent interest. We refer the reader to Section 3.4 for an elaboration. Finally it should be noted that non-black-box stegosystems can be much more efficient—see [10, 17, 13, 14].

2 Definitions

2.1 Steganography

The definitions here are essentially those of [10]. 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) explicit in the definition.

The Channel. Let Σ be an alphabet; we call the elements of Σ *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., $Pr_{D_{s_1 s_2 \cdots s_{i-1}}}[s_i] > 0$. 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 κ,

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 κ 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 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. [10, 17, 1]). 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 is a (t,d,q,l) *passive adversary for stegosystem* S if

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 κ 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

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

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 [6] 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 [8] to the existence of one-way functions that are secure given the channel oracle. This is the minimal assumption needed for steganography [10].

Let $\mathcal{F} = \{F_{\text{seed}}\}_{\text{seed}\in\{0,1\}^*}$ 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

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

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

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

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 receiver, 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 Σ. 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 Σ), 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 Σ 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 [7] (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 Σ, we write $\vec{D} = D_1 \times D_2 \times \ldots$ to denote the channel that, on history length i, outputs a uniformly random element of D_i; if $|D_1| = |D_2| = \ldots = 2^h$ then we say that $\vec{D}$ is a *flat h-channel*. Normally, one would think of the channel sampling oracle for $\vec{D}$ 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:

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

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 $\vec{D}$.

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 $N \in \mathbb{N}$:

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

Proof. Simple combinatorics show that the number of different sequences t that have weight at most N (and hence the number of messages that have encodings of weight at most N) is at most $\binom{N}{l}$: indeed, it is simply the number of positive integer solutions to $x_1 + \ldots + x_l \leq N$, 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 $\binom{N}{l} < \left(\frac{Ne}{l}\right)^l$. □

Our lower bound applies when a stegosystem is used to encode messages drawn uniformly from bit strings of equal length. It can easily be extended to messages drawn from a uniform distribution on any set.

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 $\vec{D}$ is a flat h-channel chosen uniformly at random. In the following lemma we 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.

Before stating the lemma, define the set $E = \{(\vec{D}, \mathcal{S}) \mid \vec{D} = D_1 \times D_2 \times \dots \text{ is a flat } h\text{-channel}; s_{i,j} \in D_i\}$. For $y = y_1 \dots y_l \in \Sigma^*$, we will use notation $y \in \vec{D}$ to mean that y_i is in the support of D_i. For an algorithm F we define $Q(F)$ to be the set of responses to the oracle queries of F; and $Q_i \subseteq Q$ to be the set of responses to queries with input i.

Lemma 2. *Consider any (possibly randomized) procedure F that is given oracle access to a random flat h-channel $\vec{D}$, whose goal is to output an element from the support of $\vec{D}$. Provided that h is sufficiently smaller than $\log S$, if F outputs something it did not get from the oracle, then its probability of success is low.*

More precisely, let $F^{\mathcal{S}} : \{0,1\}^* \to \Sigma^*$ (the input to F is its randomness). Define the following two events, each a subset of $E \times \{0,1\}^*$:

- *non queried:* $Nq = \{((\vec{D}, \mathcal{S}), r) \mid (\exists i) F^{\mathcal{S}}(r)_i \notin Q_i(F^{\mathcal{S}}(r)) \ \wedge \ |F^{\mathcal{S}}(r)| = l\}$
- *in support:* $Ins = \{((\vec{D}, \mathcal{S}), r) \mid F^{\mathcal{S}}(r) \in \vec{D} \ \wedge \ |F^{\mathcal{S}}(r)| = l\}$

Then:

$$\Pr_{(\vec{D},\mathcal{S}) \in E, r \in \{0,1\}^*}[Ins \ \wedge \ Nq] < \frac{H}{S}$$

Proof. Let $A_i = \{(\vec{D}, \mathcal{S}, r) \mid (F^{\mathcal{S}}(r))_i \notin Q_i\}$, for $i \in \{1, \dots, l\}$. Now $B_i = A_i \setminus (A_1 \cup \dots A_{i-1})$ is the set of all $(\vec{D}, \mathcal{S}, r)$ for which the i-th coordinate $F^{\mathcal{S}}(r)$ is not sampled, but coordinates $1, \dots, i-1$ are. Clearly, B_i are disjoint and $\bigcup_{i=1}^{l} B_i = A$ where $A = \bigcup_{i=1}^{l} A_i$. Now the probability we are interested in can be upper bounded by

$$\left(\sum_{1 \le i \le l,} \underbrace{\Pr[y = F^{\mathcal{S}}(r) \in D_1 \times \dots \times (D_i \setminus Q_i) \times \dots \times D_l \mid (\vec{D}, \mathcal{S}, r) \in B_i]}_{(\star)} \cdot \Pr[B_i] \right) \cdot \frac{1}{\Pr[A]}$$

To bound $(\star)$ for some i, fix everything but the i-th component of $\vec{D}$:

$$\Pr_{\text{random flat } h\text{-channel}_{D_i}}[F^{\mathcal{S}}(r) \in D_1 \times \dots \times (D_i \setminus Q_i) \times \dots \times D_l] = \frac{\binom{S-|Q_i|-1}{H-|Q_i|-1}}{\binom{S-|Q_i|}{H-|Q_i|}} = \frac{H-|Q_i|}{S-|Q_i|} \le \frac{H}{S}.$$

Since this quantity is independent of other coordinates of $\vec{D}$, as well as of s and r, we have that $(\star) \le \frac{H}{S}$. Finally, since the disjoint union over of B_i forms the whole space of interest to us, we have that

$$\sum_{1 \le i \le l} (\star) \cdot \frac{\Pr[B_i]}{\Pr[A]} \le \frac{H}{S}.$$

□

3.3 Lower Bound for Unbounded Adversary

Using Lemma 2, it is easily shown that, if the stegoencoder has insecurity ϵ, then it cannot output something it did not receive as response to a query with probability higher than $\epsilon/(1 - H/S)$. However, this holds only if we assume that the adversary can test whether s_i is in the support of D_i. This is not possible if D_i is completely random and the adversary's description is small compared to $S = |\Sigma|$. However, it works if the adversary is unbounded, and serves as a useful warm-up. It leads to the following theorem.

Theorem 1. *Let (SE, SD) be a black-box stegosystem with insecurity ϵ against an adversary who has an oracle for testing membership in the support of $\mathcal{C}$, unreliability ρ and rate w for an alphabet Σ 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. We prove Theorem 1 by combining Lemmas 1 and 2 (see the paragraph before Lemma 2 for some additional notation).

We define the following events, which are all subsets of $E \times \{0,1\}^* \times \{0,1\}^{lw} \times \{0,1\}^*$ (below v denotes the randomness of SE):

- "SE makes few queries to encode m under K": $Few = \{\vec{D}, \mathcal{S}, K, m, v \mid SE^{\mathcal{S}}(K, m; v))$ makes at most N queries$\}$
- "SE outputs a correct encoding of m under K": $Corr = \{\vec{D}, \mathcal{S}, K, m, v \mid SD(K, SE^{\mathcal{S}}(K, m; v)) = m\}$
- "m has an encoding t under K, and this encoding has low weight": $Lw = \{\vec{D}, \mathcal{S}, K, m, v(\exists t) \mid SD(K, t) = m \ \wedge\ W(t, \mathcal{S}) \leq N\}$
- Ins and Nq as in Lemma 2, but as subsets of $E \times \{0,1\}^*\{0,1\}^{lw} \times \{0,1\}^\infty$

Suppose that SE outputs a correct encoding of a message m. In that case, the probability that it made at most N queries to the channel, is upper bounded by the probability that: (i) there exists an encoding of m of weight at most N, or (ii) SE output something it did not query. In other words,

$$\Pr[Few \mid Corr] \leq \Pr[Lw \mid Corr] + \Pr[Nq \mid Corr].$$

Now we have

$$\begin{aligned}
\Pr[Few] &= \Pr[Few \cap Corr] + \Pr[Few \cap \overline{Corr}] \\
&\leq \Pr[Few \cap Corr] + \Pr[\overline{Corr}] \\
&= \Pr[Few \mid Corr] \cdot \Pr[Corr] + \Pr[\overline{Corr}] \\
&\leq (\Pr[Lw \mid Corr] + \Pr[Nq \mid Corr]) \cdot \Pr[Corr] + \Pr[\overline{Corr}] \\
&= \Pr[Lw \cap Corr] + \Pr[Nq \cap Corr] + \Pr[\overline{Corr}] \\
&\leq \Pr[Lw] + \Pr[Nq] + \Pr[\overline{Corr}].
\end{aligned}$$

But because the reliability is at least $1-\rho$, we have that

$$\Pr[Few] \leq \Pr[Lw] + \Pr[Nq] + \rho. \tag{1}$$

Now notice that, if the encoder outputs something not in $\vec{D}$, then it must have not queried it, i.e. $\overline{Ins} \subseteq Nq$. Because of this, we have that $\Pr[\overline{Ins} \mid Nq] = \Pr[\overline{Ins}]/\Pr[Nq]$ and so $\Pr[Nq] = \Pr[\overline{Ins}]/\Pr[\overline{Ins} \mid Nq]$, but because the insecurity is ϵ, it holds that

$$\Pr[Nq] \leq \epsilon/\Pr[\overline{Ins} \mid Nq].$$

By Lemma 2 we know that $\Pr[Ins \mid Nq] < H/S$ and so

$$\Pr[Nq] \leq \frac{\epsilon}{1-H/S}. \tag{2}$$

Finally, by Lemma 1 we have

$$\Pr[Lw] \leq \left(\frac{Ne}{l2^w}\right)^l. \tag{3}$$

Now by combining (1), (2) and (3) we get that

$$\Pr[Few] \leq \left(\frac{Ne}{l2^w}\right)^l + \rho + \frac{\epsilon}{1-H/S}.$$

Note that the probability is taken, in particular, over a random choice of $\vec{D}$. Therefore, it holds for at least one flat h-channel.

Let random variable q be equal to the number of queries made by SE to encode m under K. Then, letting $d = l2^w/e$ and $c = 1 - \rho - \frac{\epsilon}{1-H/S}$, we get

$$\mathrm{E}[q] \;=\; \sum_{N\geq 0} \Pr[q > N] \geq \sum_{N=0}^{\lceil d\rceil -1} c - \left(\frac{N}{d}\right)^l \geq \sum_{N=0}^{\lceil d\rceil -1} c - \frac{N}{d} = c\lceil d\rceil - \frac{(\lceil d\rceil -1)\lceil d\rceil}{2d} \geq \left(c - \frac{1}{2}\right)\lceil d\rceil\,.$$

The expected number of queries per document sent is $(\mathrm{E}[q])/l$ and so it is greater than $(\frac{1}{2} - \rho - \frac{\epsilon}{1-H/S})(2^w/e)$. □

3.4 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 ϵ is low unless the adversary's description size and running time are feasible ("feasible," 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[2] in the sense of [7], 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 d, insecurity $\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\mathcal{C}}(t, d, q, l)$, unreliability ρ, rate w and running time τ for an alphabet Σ 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

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

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

Proof. The main challenge lies in formulating the analogue of Lemma 2 under computational restrictions. Lemma 2 relies on: (i) the inability of the encoder to predict the behaviour of the channel (because the channel is random) and (ii) the ability of the adversary to test if a given string is in the support of the channel (which the adversary has because it is unbounded). We need to mimic this in the computationally bounded case. We do so by constructing a channel whose support (i) appears random to a bounded encoder, but (ii) has an efficient test of membership that the adversary can perform given only the short advice. As already mentioned, we wish to replace a pseudorandom channel with a random one and give the short pseudorandom seed to the adversary, while keeping the min-entropy guarantee truthful.

Given the work of [7], it would be straightforward to specify the channel as a random object (random subset D of Σ of size H) admitting two types of queries: "sample" and "test membership." But another small wrinkle is that a pseudorandom implementation of such an object would also replace random sampling with pseudorandom sampling, whereas in a stegosystem the encoder is guaranteed a truly random sample from D (indeed, without such a guarantee, the min-entropy guarantee is no longer meaningful). Therefore, we need to construct a slightly different random object, implement it pseudorandomly, and add random sampling on top of it. We specify the random object as follows. Recall that $S = |\Sigma|$, h is the min-entropy, and $H = 2^h$.

Definition 3 (Specification of a flat h-channel). Let M_ω be a probabilistic Turing machine with an infinite random tape ω. On input $(S, H, t, a, b) \in \mathbb{N}^5$, M_ω does the following:

- divides ω into consecutive substrings $y_1, y_2, \ldots$ of length S each;

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

- identifies among them the substrings that have exactly H ones; let y be the t-th such substring (with probability one there are infinitely many such substrings, of course);
- returns the number of ones in y between, and including, positions a and b in y (on ill-formed inputs, it returns some error symbol)

In what way does $M = M_\omega$ specify a flat h-channel? To see that, identify Σ with $\{1, \ldots, S\}$, and let D_t be the subset of Σ indicated by the ones in y. Then D_t has cardinality H and testing membership in D_t can be realized using a single query to M:

```
insupp^M(t, a):
    return M(S, H, t, a, a)
```

Obviously, D_t are selected uniformly at random, and independently of each other. Thus, this object specifies the correct channel and allows membership testing.

We now use this object to allow for random sampling of D_t. Outputting a random element of D_t can be realized via $\log S$ queries to M, using the following procedure (essentially, binary search):

```
rndelt^M(t):
    return random-element-in-range^M(S, H, t, 1, S)

random-element-in-range^M(S, H, t, a, b):
    if a = b then return a and terminate
    m ← ⌊(a + b)/2⌋
    total ← M(S, H, t, a, b)
    left ← M(S, H, t, a, m)
    r ←R {1, ..., total}
    if r ≤ left then
        random-element-in-range^M(S, H, t, a, m)
    else
        random-element-in-range^M(S, H, t, m + 1, b)
```

We can implement this random object pseudorandomly using techniques of [7][3]. The supports $D_1, D_2, \ldots$ will be selected pseudorandomly, and allow for efficient membership testing given short advice (pseudorandom seed, essentially) — but they will still have the requisite min-entropy h. Furthermore the sampling procedure $\texttt{rndelt}(t)$ will still select a truly random element from D_t. Therefore, it is valid to expect proper performance of the encoder on the channel specified by an implementation. However, the adversary will be able to test membership in the channel using only the short seed used by the pseudorandom implementation.

Let us now introduce some notation that will allow us to state the claim about existence of pseudo-implementations of flat h-channels.

[3]In fact, we only slightly modify one of their constructions, namely that of *interval sums of random boolean functions*. The authors [7] give a construction of a truthful pseudo-implementation of a random object determined by a random boolean function $f : \{1, \ldots, S\} \to \{0, 1\}$ that accepts queries of the form $(a, b) \in \mathbb{N}^2$ and answers $\sum_{i=a}^{b} f(i)$. Roughly, their construction is as follows. Imagine a full binary tree of depth $\log S$, whose leaves contain values $f(1), f(2), \ldots, f(S)$. Any other node in the tree contains the sum of leaves reachable from it. Given access to such tree, we can compute any sum $f(a)+f(a+1)+\ldots+f(b)$ in time proportional to $\log S$. Moreover, such trees need not be stored fully, but can be evaluated dynamically. The value in the root (i.e. the sum of all leaves) has binomial distribution, and can be filled in pseudorandomly. Other nodes have more complex distributions, but can too be filled in pseudorandomly and consistently, so that they contain the sums of their leaves. We refer the reader to [7] for details. The modification that we make, is simply fixing the value in the root to H, so that $f(1) + f(2) + \ldots + f(S) = H$.

Consider MI_ω, an implementation of a flat h-channel with random tape ω. How should we denote that "a machine A has access to a channel given by MI_ω"? For sake of consistency with the computationally unbounded case, we do not let A interact directly with $\texttt{rndelt}^{MI_\omega}$, but rather give it access to a fixed string — a draw sequence. Similarly to the computationally unbounded case, we define:

$$DPR_t^\omega = \{a \mid \texttt{insupp}^{MI_\omega}(t, a) = 1\}$$

$$\overrightarrow{DPR^\omega} = DPR_1^\omega \times DPR_2^\omega \times \dots$$

$$EPR_n = \{(\omega, \mathcal{S}) \mid |\omega| = n, s_{i,j} \in DPR_i^\omega\}\,.$$

Like for the unbounded case, we write $y \in \overrightarrow{DPR^\omega}$ to mean $y_1 \in DPR_1^\omega$, …, $y_{|y|} \in DPR_{|y|}^\omega$; and, for an algorithm F, we define $Q(F)$ to be the set of responses to the oracle queries of F, and $Q_i \subseteq Q$ to be the set of responses to queries with input i.

Clearly, for given n, picking at random $(\omega, \mathcal{S}) \in EPR_n$ amounts to picking at random a channel given by MI, and then a random draw sequence from that channel. Hence, to describe an experiment in which A takes samples from a random channel given by MI_ω ($|\omega| = n$) and where we are interested in A's outcome being equal to 1, we write $\Pr_{(\omega,\mathcal{S})\in EPR_n}[A^s(x) = 1]$.

In the following claims, we assume existence of a family of pseudorandom functions $\mathcal{F}$ with insecurity $\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(t, d, q)$ (recall **InSec** is a bound on the distinguishing advantage of any adversary running in time at most t of description size at most d making at most q queries). To simplify notation, we will omit the adversary's description size d (it is not important, because this PRF needs to be secure only in the standard model, not in the model with channel oracles; throughout, the description size d will be equal to the description size of the stegosystem plus $O(1)$, because the stegosystem ultimately will be the distinguisher) and upperbound q by t. We will then write $\iota_{PRF}(n, t)$ instead of $\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(t, d, q)$

The following claim establishes that our pseudorandom channels can be implemented, and follows from [7].

Claim 1. *There is a polynomial p and a machine MI_ω with random tape ω of length n which runs in time polynomial in n and $\log S$, such that: for oracle machine A_r with input 1^k and random tape r running in time τ, and for any $S, H \in \mathbb{N}$, $H < S$*

$$\left|\Pr_{(\omega,\mathcal{S})\in E, r\in\{0,1\}^\tau}[A_r^s(1^k) = 1] - \Pr_{(\omega,\mathcal{S})\in EPR_n, r\in\{0,1\}^\tau}[A_r^s(1^k) = 1]\right| < \iota_{PRF}(n, p(\tau)) + \tau \cdot 2^{-n}\,,$$

Note that the second argument to ι_{PRF} does not depend on S (it depends on S only to the extent that τ does, whose dependence can be logarithmic); this is important, because we want to keep the second argument to ι_{PRF} as low a possible so that ι_{PRF} is as low as possible.

Finally we are ready to state the lemma crucial for establishing the lower bound on the number of queries in the computationally bounded case. Speaking loosely, we prove: any sampler that with high probability outputs elements of the support of a pseudorandom flat h-channel, must with high probability output only what it had queried. More precisely, for a sampler F_r with random tape r and running time τ we define the following two families of events, indexed by n, the security of the pseudo-implementation of the channel.

- $InsPR_n = \{((\omega, \mathcal{S}), r) \in EPR_n \times \{0,1\}^{\tau(k)} \mid y = F_r^{\mathcal{S}}(1^k), (\forall t) y_t \in DPR_t^\omega\}$; this event isolates the points in the probability space on which the sampler successfully outputs elements from the support of the channel

- $NqPR_n = \{((\omega, \mathcal{S}), r) \in EPR_n \times \{0,1\}^{\tau(k)} \mid y = F_r^{\mathcal{S}}(1^k), (\exists t) y_t \notin Q_t(F_r^{\mathcal{S}}(1^k))\}$; this event isolates the points on which the sampler outputs something it had not queried.

We show that high probability of $InsPR_n$ implies low probability of $NqPR_n$. Formal statement of the lemma follows. To simplify notation, let $R = 1/(1 - H/S)$.

Lemma 3. *There exists a polynomial p with the following property. Let $F_r^{\mathcal{S}}$ be a probabilistic sampler with running time τ and random tape r, and let $S, H \in \mathbb{N}$, $H < S$. If $\Pr[\overline{\mathrm{InsPR}_n}] < \epsilon(n)$, then:*

$$\Pr[\mathrm{NqPR}_n] < R\epsilon(n) + (R+1)(\iota_{PRF}(n, p(\tau)) + 2^{-n}) .$$

Proof. Let Ins and Nq be as in the proof of Theorem 1. Note that it is easy to construct an efficient oracle machine that: (i) given oracle MI_ω, runs the sampler and then tests if its output is in DPR_t^ω; (ii) given oracle M, runs the sampler and tests if its output is in $\vec{D}$ determined by M. This machine simply selects at random ω, then simulates F and answers its queries using `rndelt` with the appropriate oracle. To test that the output is in the support, `insupp` with the same oracle is used. Depending on the oracle, this machine either succeeds on $InsPR_n$, or on Ins. A similar oracle machine for $NqPR$ and Nq can be constructed.

By the previous observation and Claim 1, we have that for some polynomial p_1:

$$|\Pr[InsPR_n] - \Pr[Ins]| < \iota_{PRF}(n, p_1(\tau)) + 2^{-n}.$$

Therefore $\Pr[\overline{Ins}] < \epsilon(n) + \iota_{PRF}(n, p_1(\tau)) + 2^{-n}$. It now follows[4] that $\Pr[Nq] < R(\epsilon(n) + \iota_{PRF}(n, p_1(tau)) + 2^{-n})$. Applying Claim 1 again, and taking heed of the observation from the beginning of this proof, we get that for some polynomial p_2, $|\Pr[NqPR_n] - \Pr[Nq]| < \iota_{PRF}(n, p_2(\tau)) + 2^{-n}$ and so

$$\Pr[NqPR_n] < R(\epsilon(n) + \iota_{PRF}(n, p_1(\tau)) + \iota_{PRF}(n, p_2(\tau)) + (1+R)2^{-n} .$$

Now let $p \geq \max(p_1, p_2)$. □

We are now ready to prove Theorem 2. Let κ be the security parameter for the stegosystem, and let τ be the running times of the stegoencoder and stegodecoder combined, and let v denote the randomness used by the encoder. The proof is similar to that of Theorem 1. For $n \in \mathbb{N}$ we define the following events, all of them being subsets of $EPR_n \times \{0,1\}^\kappa \times \{0,1\}^{wk} \times \{0,1\}^\tau$:

- $Few_n = \{|Q(SE^s(K, m; v))| \leq N\}$
- $Corr_n = \{SD(K, SE^s(K, m; v)) = m\}$
- $Lw_n = \{(\exists t) SD(K, t) = m \ \wedge \ W(s, t) \leq N\}$
- $Ins_n = \{SE^s(K, m; v) \in \overrightarrow{DPR^\omega}\}$
- $Nq_n = \{(\exists i)(SE^s(K, m; v))_i \notin Q(SE^s(K, m, v))\}$

[4] The argument is similar to the one found in the proof of Theorem 1. First note that $\overline{Ins} \subseteq Nq$. Hence, $\Pr[Nq] = \Pr[\overline{Ins}]/\Pr[\overline{Ins} \mid Nq]$. By Lemma 2 we have that $\Pr[Ins \mid Nq] \leq H/S$, and we saw that $\Pr[\overline{Ins}] \leq \epsilon(n) + \iota_{PRF}(n, p_1(\tau))$

Just like in the proof of 1, one argues that $\Pr[Few_n] \leq \Pr[Lw_n] + \Pr[Nq_n] + \Pr[\overline{Corr_n}]$ and then that $\Pr[\overline{Corr_n}] < \rho$ and $\Pr[Lw_n] < (Ne/l2^w)^l$. It follows that $p_n \leq (Ne/l2^w)^l + \rho + \Pr[Nq_n]$. It is left to argue that $\Pr[Nq_n] < \nu(n) + \epsilon R$.

Consider an adversary against our stegosystem that contains ω as part of its description, gives its oracle a random message to encode, and then tests if the output is in $\overrightarrow{DPR^\omega}$. It can be implemented to run in $q(\log S, n, l)$ steps for some polynomial q [5], and has description size $n+c$ for some constant c. Hence its probability of detecting an stegoencoder output that is not in $\overrightarrow{DPR^\omega}$ cannot be more than $\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\overrightarrow{DPR^\omega}}(q(\log S, n), n+c, 1, lw)$. In other words,

$$\Pr[\overline{Ins_n}] \leq \mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\overrightarrow{DPR^\omega}}(q(\log S, n), n+c, 1, lw)\,.$$

By Lemma 3 we get

$$\Pr[Nq_n] \leq R\mathbf{InSec}^{\mathrm{SS}}_{S(\kappa),\overrightarrow{DPR^\omega}}(q(\log S, n), n+c, 1, lw) + (R+1)(\iota_{PRF}(n, p(\tau)) + 2^{-n})$$

for some polynomials p, q and constant c.

Finally, to compute a bound on the expected value, we apply the same method as in the proof of Theorem 1. □

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 [7] 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 [7]. 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[6] 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 κ (the length of the PRF key K), it depends on the rate parameter w. Because it is

[5] For each y_t it needs to test membership to DPR^ω_t, i.e. $\mathtt{insupp}^{MI_\omega}(t, y_t)$. Since MI runs in time polynomial in n and $\log S$, then for given ω, $\mathtt{insupp}^{MI_\omega}$ can be implemented to run for $\mathrm{poly}(\log S, n)$ steps.

[6] 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.

stateful, both encoder and decoder take a counter ctr as input.

Our encoder is similar to the rejection-sampler-based encoder of [10] 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 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 [11, 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) [15, 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 [2] at the expense of introducing potential false collisions and thus further decreasing reliability. An analysis utilizing the bounds of [3] (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 Σ 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 [10]). 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 $m = m_1 \dots m_l$, where $|m_i| = w$
  `for` $i \leftarrow 1$ `to` l:
    $j \leftarrow 0$
    `repeat` :
      $j \leftarrow j + 1$
      $s_{i,j} \leftarrow M(\mathcal{H})$
    `until` $F_K(s_{i,j}) = m_i$ `or` $j = k$
    $s_i \leftarrow s_{i,j}$; $\mathcal{H} \leftarrow \mathcal{H}||s_i$
  `output` $s = s_1 s_2 \dots s_l$

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

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

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

More precisely,

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

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).

The encoders SE_2, SE_3, and SE_4 are specified in Figure 1. 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 follow.

For ease of notation, we will denote 2^{-h} (the upper bound on the probability of elements of $D_{\mathcal{H}}$) by p and 2^w by R for the rest of this proof.

The following proposition serves as a warm-up for the proof of the Lemma 4, which follows it.

$SE_2(K, w, k, m_1 \dots m_l, \mathcal{H})$:
$Q \leftarrow \emptyset$
`for` $i \leftarrow 1$ `to` l:
  $j \leftarrow 0$
  `repeat` :
    $j \leftarrow j + 1$
    $s_{i,j} \leftarrow M(\mathcal{H})$
    `if` $s_{i,j} \in Q$ `or` $j = k + 1$ `then`
      `abort` and `output` "Fail"
    $Q \leftarrow Q \cup \{s_{i,j}\}$
  `until` $G(s_{i,j}) = m_i$
  $s_i \leftarrow s_{i,j}$; $\mathcal{H} \leftarrow \mathcal{H}||s_i$
`output` $s = s_1 s_2 \dots s_l$

$SE_3(K, w, k, m_1 \dots m_l, \mathcal{H})$:
$Q \leftarrow \emptyset$
`for` $i \leftarrow 1$ `to` l:
  $j \leftarrow 0$
  `repeat` :
    $j \leftarrow j + 1$
    $s_{i,j} \leftarrow M(\mathcal{H})$
    `if` $s_{i,j} \in Q$ `or` $j = k + 1$ `then`
      `abort` and `output` "Fail"
    $Q \leftarrow Q \cup \{s_{i,j}\}$
    Pick $c \in_R \{0,1\}^w$
  `until` $c = m_i$
  $s_i \leftarrow s_{i,j}$; $\mathcal{H} \leftarrow \mathcal{H}||s_i$
`output` $s = s_1 s_2 \dots s_l$

$SE_4(K, w, k, m_1 \dots m_l, \mathcal{H})$:
`for` $i \leftarrow 1$ `to` l:
  $j \leftarrow 0$
  `repeat` :
    $j \leftarrow j + 1$
    $s_{i,j} \leftarrow M(\mathcal{H})$
    Pick $c \in_R \{0,1\}^w$
  `until` $c = m_i$
  $s_i \leftarrow s_{i,j}$; $\mathcal{H} \leftarrow \mathcal{H}||s_i$
`output` $s = s_1 s_2 \dots s_l$

Figure 1: "Encoders" SE_2, SE_3 and SE_4 used in the proof of Theorem 4

Proposition 1. *The statistical difference between the output distributions of SE_1 and SE_2 for a w-bit hiddentext message $m \in \{0,1\}^w$ is at most $2p/(R-1)^2 + 2e^{-k/R}$. That is,*

$$\sum_{\forall s \in \Sigma} \left| \Pr_{G,M}[SE_1(K, w, k, m, \mathcal{H}) \to s] - \Pr_{G,M}[SE_2(K, w, k, m, \mathcal{H}) \to s] \right| < 2p(R-1)^2 + 2e^{-k/R}\,.$$

Proof. Consider the probability that SE_2 outputs "Fail" while trying to encode some $m \in \{0,1\}^w$. This happens for one of two reasons. First, if after k attempts to find $s_{i,j}$ such that $G(s_{i,j}) = m$, no such $s_{i,j}$ has been drawn. Second, if before SE_2 finds a satisfactory $s_{i,j}$, the same value is return twice by M: in other words, there is been a collision between the unsuccessful covertext documents.

Let E_1 denote the event that one of these situations has occurred and n_1 denote the value of j at which the event occurs. Then

$$\begin{aligned}
\Pr[E_1] &\leq \left(\frac{R-1}{R}\right)^2 p + \left(\frac{R-1}{R}\right)^3 2p + \dots + \left(\frac{R-1}{R}\right)^{k-1} (k-2)p + \left(\frac{R-1}{R}\right)^k \\
&= p \sum_{n_1=2}^{k-1} \left(\frac{R-1}{R}\right)^{n_1} (n_1 - 1) + \left(\frac{R-1}{R}\right)^k \\
&< p \left(\frac{R-1}{R}\right)^2 \sum_{n_1=0}^{\infty} \left(\frac{R-1}{R}\right)^{n_1} (n_1 + 1) + \left(\frac{R-1}{R}\right)^k \\
&= p(R-1)^2 + \left(\frac{R-1}{R}\right)^k \\
&< p(R-1)^2 + e^{-k/R}\,.
\end{aligned}$$

Observe that the probability that SE_2 outputs a specific document s which is not "Fail" can only be less than the probability that SE_1 outputs the same element. Since the total decrease over all such s is at most the probability of failure from above, the total statistical difference is at most $2\Pr[E_1]$. □

Lemma 4. *The statistical difference between the output of SE_1 and SE_2 when encoding a message $m \in \{0,1\}^{lw}$ is at most*

$$p\left(l(l+1)R^2 - l(l+3)R + 2l\right) + 2l\left(1 - \frac{1}{R}\right)^k .$$

Proof. Proposition 1 deals with the case of $l = 1$. It remains to extend this line of analysis to the general case of $l > 1$. As in the proof of Proposition 1, let E_i denote the event that SE_2 outputs "Fail" while attempting to encode the ith block of m_i. Note that E_i grows with i because the set Q grows as more and more blocks are encoded. Also, let n_i denote the number of attempts used by SE_2 to encode the ith block. To simplify the analysis, we initially ignore the boundary case of failure on attempt $n_i = k$ and treat a failure on this attempt like all others. Let E'_i denote these events. Then, we have the following sequence of probabilities.

Recall that for E'_1,

$$\Pr[E'_1] < p(R-1)^2 .$$

In the harder case of E'_2,

$$\begin{aligned}
\Pr[E'_2] &= \sum_{n_1=1}^{k} \Pr[E'_2 | n_1 \text{ draws for bit 1}] \Pr[n_1 \text{ draws for bit 1}] \\
&\leq \frac{p}{R} \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \left(\frac{R-1}{R}\right)^{n_1+n_2-1} (n_1 + n_2 - 1) \\
&= \frac{p}{R} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(\sum_{n_2=1}^{k} \left(\frac{R-1}{R}\right)^{n_2} (n_2 - 1) + n_1 \sum_{n_2=1}^{k} \left(\frac{R-1}{R}\right)^{n_2} \right) \\
&< \frac{p}{R} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(\Pr[E'_1]/p + n_1(R-1)\right) \\
&< \frac{p}{R} \left(R \Pr[E'_1]/p \ + R^2(R-1)\right) \\
&= p\left((R-1)^2 + R(R-1)\right) \\
&= p(2R-1)(R-1) .
\end{aligned}$$

Similarly for E'_3,

$$\begin{aligned}
\Pr[E'_3] &\leq \frac{p}{R^2} \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \sum_{n_3=1}^{k} \left(\frac{R-1}{R}\right)^{n_1+n_2+n_3-2} (n_1 + n_2 + n_3 - 1) \\
&= \frac{p}{R^2} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(R \Pr[E'_2]/p + n_1 \sum_{n_2=1}^{k} \left(\frac{R-1}{R}\right)^{n_2-1} \sum_{n_3=1}^{k} \left(\frac{R-1}{R}\right)^{n_3} \right) \\
&< \frac{p}{R^2} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(R \Pr[E'_2]/p + n_1 R(R-1)\right) \\
&< \frac{p}{R^2} \left(R^2 \Pr[E'_2]/p + R^3(R-1)\right) \\
&= p(3R-1)(R-1) .
\end{aligned}$$

In general, for E'_i we have the recurrence,

$$\begin{aligned}\Pr[E'_i] &\leq \frac{p}{R^{i-1}} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(R^{i-2}\Pr[E'_2]/p + n_1 R^{i-2}(R-1)\right) \\ &< \Pr[E'_{i-1}] + pR(R-1)\,,\end{aligned}$$

which when solved yields,

$$\Pr[E'_i] < p(iR-1)(R-1)\,.$$

Now summing up the probability of failure for each of the w-bit blocks of hiddentext gives,

$$\begin{aligned}\sum_{i=1}^{l} \Pr[E'_i] &< p(R-1)\sum_{i=1}^{l}(iR-1) \\ &= p(R-1)\left(R\sum_{i=1}^{l} i - \sum_{i=1}^{l} 1\right) \\ &= p(R-1)\left(\frac{Rl(l+1)}{2} - l\right) \\ &= p\left(\left(\frac{R^2}{2}\right)(l+1)l - \left(\frac{R}{2}\right)(l+3)l + l\right)\,.\end{aligned}$$

Next, we compute the probability of the event that the encoding of block m_i fails because there were k unsuccessful attempts to find a string of n covertexts which evaluates to m_i under G, given that no collisions occurred so far. Call this event $\hat{E}_i$. Then

$$\Pr[\hat{E}_i] < \left(\frac{R-1}{R}\right)^k\,.$$

$$\begin{aligned}\Pr[\hat{E}_1] &< \left(\frac{R-1}{R}\right)^k\,, \\ \Pr[\hat{E}_2] &< \frac{1}{R}\sum_{n_1=1}^{k}\left(\frac{R-1}{R}\right)^{n_1+k-1} = \frac{\Pr[\hat{E}_1]}{R}\sum_{n_1=1}^{k}\left(\frac{R-1}{R}\right)^{n_1-1} < \Pr[\hat{E}_1]\,, \\ \Pr[\hat{E}_3] &< \frac{1}{R^2}\sum_{n_1=1}^{k}\sum_{n_2=1}^{k}\left(\frac{R-1}{R}\right)^{n_1+n_2+k-2} = \frac{1}{R^2}\Pr[\hat{E}_1]\sum_{n_1=1}^{k}\sum_{n_1=1}^{k}\left(\frac{R-1}{R}\right)^{n_1+n_2-2} < \Pr[\hat{E}_1]\,,\end{aligned}$$

and in general,

$$\Pr[\hat{E}_i] < \frac{1}{R^{i-1}}\sum_{n_1=1}^{k}\sum_{n_2=1}^{k}\cdots\sum_{n_{i-1}=1}^{k}\left(\frac{R-1}{R}\right)^{n_1+n_2+\cdots+n_{i-1}+k-(i-1)} < \Pr[\hat{E}_1]\,.$$

So, the sum of the probabilities of the $\hat{E}_i$s for i from 1 to l is just $l\Pr[\hat{E}_1] = l\left(\frac{R-1}{R}\right)^k$.

Finally we compute the total probability of failure which is at most the sum of the E'_i and $\hat{E}_i$ events. That is, the probability that SE_2 outputs "Fail" while encoding any of the l w-bit blocks of m_i of m is at most

$$\begin{aligned}\sum_{i=1}^{l} \Pr[E_i] &< \sum_{i=1}^{l} \Pr[E'_i] + \Pr[\hat{E}_i] \\ &< p\left(\left(\frac{R^2}{2}\right)(l+1)l - \left(\frac{R}{2}\right)(l+3)l + l\right) + l\left(\frac{R-1}{R}\right)^k .\end{aligned}$$

The statistical difference is at most just twice this amount. □

Lemma 5. *The statistical difference between the output distributions of SE_2 and SE_3 for a random function G and hiddentext message $m \in \{0,1\}^{lw}$ is zero.*

Proof. Both SE_2 and SE_3 abort and output "Fail" whenever the encoding a block m_t fails. This occurs because either: (1) there are k unsuccessful attempts to find $s_{i,j}$ such that $G(s_{i,j}) = k$; or (2) the same document is drawn twice, i.e. there is a collision between candidate covertext documents. Hence, SE_2 evaluates G at most once on each element of Σ. So, although SE_3 ignores G and creates its own random function by flipping coins at each evaluation, since no element of Σ will be re-assigned a new value, the output distributions of SE_2 and SE_3 are identical. □

Lemma 6. *The statistical difference between the output distributions of SE_3 and SE_4 is equal to the statistical difference between the output distributions of SE_1 and SE_2 used to encode the same message.*

Proof. As Lemma 4 shows, the probability that SE_2 (and consequently SE_3 by Lemma 5) outputs "Fail" is at most

$$\left(\left(\frac{R^2}{2}\right)(l+1)l - \left(\frac{R}{2}\right)(l+3)l + l\right) + l\left(\frac{R-1}{R}\right)^k .$$

Note that SE_4 has no such element; the probabilities of each output other that "Fail" can only increase. Hence, the total statistical difference is twice the probability of "Fail." □

These three Lemmas, put together, conclude the proof of the Theorem. We can save a factor of two in the statistical difference by the following observation. Half of the statistical difference between the outputs of SE_1 and SE_2, as well as between the outputs of SE_3 and SE_4, is due to the probability of "Fail". Because neither SE_1 nor SE_4 output "Fail," the statistical difference between the distributions they produce is therefore only half of the sum of the statistical differences. □

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 $\Sigma = \{s_1 \dots s_n\}$ 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 [9]), we obtain

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

where the second to last step follows from $p_i \leq 2^{-h}$ 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

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

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

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

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

[3] 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.

[4] 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.

[5] Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. Upper and lower bounds on black-box steganography. In Joe Kilian, editor, *Second Theory of Cryptography Conference — TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 227–244. Springer-Verlag, 2005.

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

[7] 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. IEEE.

[8] 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.

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

[10] 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.

[11] 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.

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

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

[14] 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.

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

[16] 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/`.

[17] 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 ε-Biased Functions

Many stegosystems [10, 17, 1] (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 Σ 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 ε 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 [16], 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 $\sqrt{l}2^{(-h+w)/2}$.