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

February 29, 2008

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 [11]. 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 [11, 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 [11] 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 [19, 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. [11] 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 (recall that h is the channel entropy), and negligible insecurity, which is independent of the channel. (A very similar construction was independently discovered by Hopper [12, 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 lower bound involves a surprising application of the huge random objects of [8], specifically of the truthful implementation of a boolean function with interval-sum queries. The lower bound 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 $l^2 2^{-h+2w}$ 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 unreliability of 2^{-h+w} per document 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 [12, 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 [11].

Constructions. In addition to introducing the complexity-theoretic model for steganography, [11] 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 [11] 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 [11], 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 [11], 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 [11], but the analysis of the Extended Abstract was later found to be flawed by [14]. Thus, the full version of [11] 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 [11], can be made black-box through the use of extractors (such as universal hash functions) in place of unbiased functions, as shown in [19].

In addition to the two constructions of [11] described above, and independently of our work, Hopper in [12] 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 [11], 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 [12, 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 [12]. Our proof techniques are quite different than those of [12], 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 [11, 19, 15, 16].

2 Definitions

2.1 Steganography

The definitions here are essentially those of [11]. 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., $Pr_{D_{s_1 s_2 ... s_{i-1}}}[s_i] > 0$. 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 $D_{\mathcal{H}}$. A stegosystem may be designed for a particular Σ and min-entropy of $\mathcal{C}$.

Definition 1. A *black-box secret-key stegosystem* for the alphabet Σ is a pair of probabilistic polynomial time algorithms $ST = (SE, SD)$ such that, for a security parameter κ,

1. SE has access to a channel sampling oracle M for a channel $\mathcal{C}$ on Σ 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 ST with security parameter κ for a channel $\mathcal{C}$ and messages of length μ is defined as

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

Unreliability is defined as $\mathbf{UnRel}_{ST(\kappa),\mathcal{C},\mu} = 1 - \mathbf{Rel}_{ST(\kappa),\mathcal{C},\mu}$.

The Adversary. We consider only passive adversaries who mount a chosen hiddentext attack on ST (stronger adversarial models for steganography have also been considered, see e.g. [11, 19, 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 $s_1 s_2 \dots s_l$ where each s_i is drawn according to $D_{\mathcal{H}\circ s_1 s_2 \dots s_{i-1}}$.

Definition 2. W is a (t,d,q,λ) *passive adversary for stegosystem* ST 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 λ 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 ST with security parameter κ for a channel $\mathcal{C}$ is defined as

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

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

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

and security $\mathbf{Sec}$ 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 [7] 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, the underlying assumption for our constructions is the existence of pseudorandom functions that are secure given the channel oracle, which is equivalent [9] to the existence of one-way functions that are secure given the channel oracle. This is the minimal assumption needed for steganography [11].

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

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

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.

The existence of pseudorandom functions is also the underlying assumption for our lower bound; however, for the lower bound, we do not need to give the adversary access to a channel oracle (because we construct the channel). To distinguish this weaker assumption, we will omit the subscript $\mathcal{C}$ from **InSec**.

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 use the notation D_i both for the distribution and for the subset (as is often done for uniform distributions).

Let H denote the number of elements of D_i (note that $H = |D_i| = 2^h$), and let $S = |\Sigma|$. Because the encoder knows the min-entropy h of the channel, if $H = S$, then the encoder knows the channel completely (it's simply uniform on Σ). Therefore, if $H = S$, then there is no meaningful lower bound on the number of queries made by the encoder to the channel oracle, because it does not need to make any queries in order to sample from the channel. Thus, we require that $H < S$ (our bounds will depend slightly on the ratio of S to $S - H$).

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 (intuitively, every good stegoencoder should work this way, because otherwise it may output something that is not in the channel, and thus be detected). To be reliable—that is, to find a set of documents that decode to the desired message—such an encoder has to make many queries, as shown in Lemma 1. Second, we formalize the intution that a good stegoencoder should output only documents it received from the channel-sampling oracle: we show that to be secure (i.e., not output something easily detectable by the adversary), a black-box SE cannot output anything it did not receive from the oracle: if it does, it has an $1 - H/S$ chance of being detected.

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). Thus, the adversary cannot be detecting a bad stegoenconder by simply carrying a list of all the entries in D_i for each i and checking if the i-th document sent by the stegoencoder is in D_i, because that would make the adversary's description too long.

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 [8] (truthfulness is important because min-entropy has to be high unconditionally).

3.1 Lower Bound When Only Query Results Are Output

If $D_1, D_2, \ldots$ are subsets of Σ, then 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.* We will consider flat h-channels.

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. However, from the point of view of the stegoencoder, it does not matter if the choice was made by the oracle in response to the query, or before the query was even made. It will be easier for us to think of the oracle as having already made and written down countably many samples from each D_i. We will denote the jth sample from D_i by $s_{i,j}$. Thus, suppose the oracle has already chosen

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

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 $\mathcal{S}$ 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 $\vec{D}$.

Denote the stegoencoder's output by $SE^{\mathcal{S}}(K, m, \mathcal{H}) = t = t_1 t_2 \ldots t_l$, where $t_i \in \Sigma$. Because we assume in this section that the stegoencoder outputs only documents it got from the channel oracle, 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: $W(t,\mathcal{S}) = \sum_{i=1}^{l} \min\{j \mid s_{i,j} = t_i\}$. In the next lemma, we prove (by looking at the *decoder*) that for any $\mathcal{S}$, most messages have high weight, i.e., must take many queries to encode.

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 $j_1 + \ldots + j_l \leq N$, 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 $j_i - 1$), 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 $\binom{N}{l} < \left(\frac{Ne}{l}\right)^l$ (which is a standard combinatorics fact and follows from $k! \geq (k/e)^k$, which in turn follows by induction on k from $e > (1 + 1/k)^k$). □

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. If the messages are not drawn from a uniform distribution, then, in principle, they can be compressed before transmission, thus requiring less work on the part of the stegoencoder. We do not provide a lower bound in such a case, because any such lower bound would depend on the compressibility of the message source.

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, or else it has a high probability of outputting something not in the channel. Assume $\vec{D}$ is a flat h-channel chosen uniformly at random. For $t = t_1 \ldots t_l \in \Sigma^*$, let $t \in \vec{D}$ denote that t_i is in D_i for each i. In the following lemma we demonstrate that, if the encoder's output t contains a document that it did not receive as a response to a query, the chances that $t \in \vec{D}$ are at most H/S.

Before stating the lemma, we define the set E of all possible flat h-channels and draw sequences consistent with them: $E = \{(\vec{D}, \mathcal{S}) \mid s_{i,j} \in D_i\}$. We will be taking probabilities over E. Strictly speaking, E is an infinite set because we defined $\vec{D}$ to be countable and $\mathcal{S}$ to have countably many samples from each D_i. For clarity, it may be easiest to think of truncating these countable sequences to a sufficiently large value beyond which no stegoencoder will ever go, thus making E finite, and then use the uniform distribution on E. Formally, E can be defined as a product of countably many discrete probability spaces (see, e.g., [6, Section 9.6]), with uniform distribution on each.

Lemma 2. *Consider any deterministic procedure A that is given oracle access to a random flat h-channel $\vec{D}$ and outputs $t = t_1t_2 \ldots t_l \in \Sigma^*$ (think of A as the stegoencoder running on some input key, message, channel history, and fixed randomness). Provided that h is sufficiently smaller than $\log S$, if A outputs something it did not get from the oracle, then the probability $t \in \vec{D}$ is low.*

More precisely, let Q_i be the set of responses A received to its queries from the ith channel D_i. Define the following two events:

- *$\underline{n}$on$\underline{q}$ueried: $Nq = \{(\vec{D}, \mathcal{S}) \in E \mid (\exists i) t_i \notin Q_i\}$*
- *$\underline{in}$ $\underline{s}$upport: $Ins = \{(\vec{D}, \mathcal{S}) \in E \mid t \in \vec{D}\}$*

Then:

$$\Pr_{(\vec{D},\mathcal{S})\in E}[Ins \ \wedge \ Nq] \leq \frac{H}{S}\,.$$

Proof. If A were always outputting just a single value ($l = 1$), the proof would be trivial: seeing some samples from a random D_1 does not help A come up with another value from D_1, and D_1 makes up only an H/S fraction of all possible outputs of A. The proof below is a generalization of this argument for $l \geq 1$, with care to avoid simply taking the union bound, which would get us lH/S instead of H/S.

Let $Nq_i = \{(\vec{D}, \mathcal{S}) \in E \mid t_1 \in Q_1, t_2 \in Q_2, \ldots, t_{i-1} \in Q_{i-1}, t_i \notin Q_i\}$ be the event t_i is the first element of the output that was not returned by the oracle as an answer to a query. Observe that and $\bigcup_i Nq_i = Nq$ and that Nq_i are disjoint events and, therefore, $\sum_i \Pr[Nq_i] = 1$. Now the probability we are interested in is

$$\Pr[Ins \ \wedge \ Nq] = \sum_i \Pr[Ins \ \wedge \ Nq_i] = \sum_i \Pr[Ins \mid Nq_i] \Pr[Nq_i]\,.$$

To bound $\Pr[Ins \mid Nq_i]$, fix any

$$\begin{aligned}\mathcal{S} = {} & s_{1,1},\ s_{1,2}, \ldots, s_{1,q_1}, \\ & s_{2,1},\ s_{2,2}, \ldots, s_{2,q_2}, \\ & \qquad \cdots \qquad ,\end{aligned}$$

such that $A^{\mathcal{S}}$ asks exactly q_1 queries from D_1, q_2 queries from D_2, ... Note that such $\mathcal{S}$ determines the behavior of A, including its output. Assume that for this $\mathcal{S}$, the event Nq_i happens. We will take the probability $\Pr[Ins \mid Nq_i]$ over a random $\vec{D}$ consistent with $\mathcal{S}$ (i.e., for which $s_{1,1}, s_{1,2}, \ldots s_{1,q_1} \in D_1, s_{2,1}, s_{2,2} \ldots s_{2,q_2} \in D_2, \ldots$). This probability can be computed simply as follows: if q_i' is the number of distinct elements in $s_{i,1}, s_{i,2}, \ldots, s_{i,q_i}$, then there are $\binom{S-q_i'}{H-q_i'}$ equally likely choices for D_i (because q_i' elements of D_i are already determined). However, for *Ins* to happen, D_i must also contain t_i, which is not among $s_{i,1}, s_{i,2}, \ldots, s_{i,q_i}$ (because we assumed Nq_i happens). The choices of $D_1, \ldots, D_{i-1}, D_{i+1}, \ldots$ do not matter. Therefore,

$$\Pr[Ins \mid Nq_i] = \frac{\binom{S-q_i'-1}{H-q_i'-1}}{\binom{S-q_i'}{H-q_i'}} = \frac{H - q_i'}{S - q_i'} \leq \frac{H}{S}\,.$$

The above probability is for *any* fixed $\mathcal{S}$ of the right length and randomly chosen $\vec{D}$ consistent with $\mathcal{S}$. Therefore, it also holds for randomly chosen $(\vec{D}, \mathcal{S}) \in E$, because the order in which $\mathcal{S}$ and $\vec{D}$ are chosen and the values in $\mathcal{S}$ beyond what A queries do not affect the probability. We thus have

$$\Pr_{(\vec{D},\mathcal{S})\in E}[Ins \ \wedge \ Nq] = \sum_i \Pr[Ins \mid Nq_i] \Pr[Nq_i] \leq \sum_i \frac{H}{S} \Pr[Nq_i] = \frac{H}{S}\,.$$

□

3.3 Lower Bound for Unbounded Adversary

We now want to tie together Lemmas 1 and 2 to come up with a lower bound on the efficiency of the stegoencoder in terms of rate, reliability and security. Note that some work is needed, because even though Lemma 1 is about reliability and Lemma 2 is about security, neither mentions the parameters **Rel** and **InSec**.

Assume, for now, that the adversary can test whether t_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 serves as a useful warm-up for the next section.) Then, 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 $\epsilon/(1 - H/S)$. This 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 for any positive integer $H < S$ 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 at most*

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

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

Note that, like Lemma 1, this theorem and Theorem 2 apply when a stegosystem is used to encode messages drawn uniformly from the distribution of all lw-bit messages (see remark following the proof of Lemma 1).

Proof. 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 <u>few</u> queries to encode m under K": $\mathit{Few} = \{\vec{D}, \mathcal{S}, K, m, v \mid SE^{\mathcal{S}}(K, m; v)$ makes at most N queries$\}$ (note that this is the event whose probability we want to bound)
- "SE outputs a <u>corr</u>ect encoding of m under K": $\mathit{Corr} = \{\vec{D}, \mathcal{S}, K, m, v \mid SD(K, SE^{\mathcal{S}}(K, m; v)) = m\}$
- "m has an encoding t under K, and this encoding has <u>low</u> weight": $\mathit{Low} = \{\vec{D}, \mathcal{S}, K, m, v(\exists t) \mid SD(K, t) = m \wedge W(t, \mathcal{S}) \leq N\}$
- Ins and Nq as in Lemma 2, but as subsets of $E \times \{0,1\}^* \times \{0,1\}^{lw} \times \{0,1\}^*$

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,

$$\Pr[\mathit{Few} \mid \mathit{Corr}] \leq \Pr[\mathit{Low} \mid \mathit{Corr}] + \Pr[\mathit{Nq} \mid \mathit{Corr}].$$

Now we have

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

Because insecurity is ϵ, $\Pr[\overline{\mathit{Ins}}] \leq \epsilon$. Hence,

$$\Pr[\mathit{Nq}] = \frac{\Pr[\overline{\mathit{Ins}} \cap \mathit{Nq}]}{\Pr[\overline{\mathit{Ins}} \mid \mathit{Nq}]} = \frac{\Pr[\overline{\mathit{Ins}}]}{\Pr[\overline{\mathit{Ins}} \mid \mathit{Nq}]} \leq \frac{\epsilon}{1 - H/S} \tag{1}$$

(the second equality follows from the fact that if the encoder outputs something not in $\vec{D}$, then it must have not queried it, i.e. $\overline{\mathit{Ins}} \subseteq \mathit{Nq}$; the inequality follows from Lemma 2).

By Lemma 1 we have

$$\Pr[\mathit{Low}] \leq \left(\frac{Ne}{l2^w}\right)^l . \tag{2}$$

Now by combining (1), (2), and the fact that $\Pr[\overline{\mathit{Corr}}] \leq \rho$ by reliability, we get that

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

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 is at least $(\frac{1}{2} - \rho - \frac{\epsilon}{1-H/S})(2^w/e)$. □

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 polynomial in the description size and running time of the stegoencode and in 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 [8], and so rely on the techniques developed therein.

The result is the following theorem, which is similar to Theorem 1, except for a small term introduced by pseudorandomness of the channel.

Theorem 2. *There exist polynomials p_1, p_2 and constants c_1, c_2 with the following property. Let $ST(\kappa)$ be a black-box stegosystem with security parameter κ, description size δ, unreliability ρ, rate w and running time τ for the alphabet $\Sigma = \{0, 1, \ldots, S-1\}$. Assume there exists a pseudorandom function family $\mathcal{F}(n)$ with insecurity $\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(t, d, q)$. Then for any positive integer $H < S$ 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*

$$\left(\frac{Ne}{l2^w}\right)^l + \rho + R\epsilon + (R+1)\left(\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(p_1(\tau, n), \delta + c_1, p_1(\tau, n)) + \tau 2^{-n}\right),$$

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

$$\frac{2^w}{e}\left(\frac{1}{2} - \rho - R\epsilon - (R+1)\left(\mathbf{InSec}^{\mathrm{PRF}}_{\mathcal{F}(n)}(p_1(\tau, n), \delta + c_1, p_1(\tau, n)) + \tau 2^{-n}\right)\right),$$

where $R = S/(S-H)$ and ϵ is the insecurity the stegosystem ST on the channel $\mathcal{C}$ against adversaries running in time $p_2(n, \log S, n)$ of description size $n + c_2$, making just one query of length lw to SE or O (i.e., $\epsilon = \mathbf{InSec}^{\mathrm{SS}}_{ST(\kappa),\mathcal{C}}(p_2(n, \log S, l), n + c_2, 1, lw)$).

Proof. The main challenge lies in formulating the analogue of Lemma 2 under computational restrictions. Lemma 2 and its use in Theorem 1 relied 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 a short advice. As already mentioned, we wish to replace a random channel with a pseudorandom one and give the short pseudorandom seed to the adversary, while keeping the min-entropy guarantee truthful.

The next few paragraphs will explain how this is done, using the techniques of huge random objects from [8]. A reader not familiar with [8] may find it easier to skip to the paragraph entitled "Properties of the Pseudorandom Flat-h Channels," where the results of this—i.e., the properties of the channel that we obtain—are summarized.

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

Specifying and Implementing the Flat-h Channel For the next few paragraphs, familiarity with [8] will be assumed. Recall that [8] requires a specification of the object that will be pseudorandomly implemented, in the form of a Turing machine with a countably infinite random tape. 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 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 five integers (S, H, i, a, b), (where $0 < H \leq S$, $i > 0$, $0 \leq a \leq b < S$), M_ω does the following:

- divides ω into consecutive substrings $y_1, y_2, \ldots$ of length S each;
- identifies among them the substrings that have exactly H ones; let y be the i-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 (positions are counted from 0 to $S-1$).

In what way does $M = M_\omega$ specify a flat h-channel? To see that, identify Σ with $\{0, \ldots, S-1\}$, and let D_i be the subset of Σ indicated by the ones in y. Then D_i has cardinality H and testing membership in D_i can be realized using a single query to M:

```
insupp^M($i$, $s$):
    return $M(S, H, i, s, s)$
```

Obviously, D_i 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_i. Outputting a random element of D_i can be realized via $\log S$ queries to M, using the following procedure (essentially, binary search):

```
rndelt^M($i$):
    return random-element-in-range^M($S, H, i, 0, S-1$)

random-element-in-range^M($S, H, i, a, b$):
    if $a = b$ then return $a$ and terminate
    $mid \leftarrow \lfloor (a+b)/2 \rfloor$
    $total \leftarrow M(S, H, i, a, b)$
    $left \leftarrow M(S, H, i, a, mid)$
    $r \xleftarrow{R} \{1, \ldots, total\}$
    if $r \leq left$ then
        random-element-in-range^M($S, H, i, a, mid$)
    else
        random-element-in-range^M($S, H, i, mid+1, b$)
```

We can implement this random object pseudorandomly using the same techniques as [8] uses for implementing random boolean functions with interval sums (see [8, Theorem 3.2]). Namely, the authors of [8] give a construction of a truthful pseudo-implementation of a random object determined by a random boolean function $f : \{0, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$ that accepts queries in the form of two n-bit integers (a, b) and answers with $\sum_{j=a}^{b} f(j)$. Roughly, their construction is as follows. Let $S = 2^n$. Imagine a full binary tree of depth n, whose leaves contain values $f(0), f(1), \ldots, f(S-1)$. Any other node in the tree contains the sum of leaves in its subtree. Given access to such tree, we can compute any sum $f(a) + f(a+1) + \ldots + f(b)$ in time proportional to n. Moreover, such trees need not be stored fully, but can be evaluated dynamically, from the root down to the leaves, as follows. 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. The construction uses a pseudorandom function to come up with the value at each node.

We need to make three modifications. First, we simply fix the value in the root to H, so that $f(0) + f(1) + \ldots + f(S-1) = H$. Second, we allow S to be not a power of 2. Third, in order to create multple distributions D_i, we add i as an input to the pseudorandom function, thus getting different (and independent-looking) randomness for each D_i.

Having made these modifications, we obtain a truthful pseudo-implementation of M. It can be used within `insupp` and `rndelt` instead of M, for efficient membership testing and truly random sampling from our pseudorandom channel.

Properties of the Pseudorandom Flat h-Channels We thus obtain that, given a short random seed ω, it is possible to create a flat h-channel that is indistinguishable from random and allows for efficient membership testing and truly random sampling given ω. To emphasize the pseudorandomness of the channel, in our notation we will use DPR insted of D and keep the seed ω explicit as a supercript. Thus, DPR_i^ω is a pseudorandom subset of Σ of size H, and the channel is denoted by $\overrightarrow{DPR}^\omega = DPR_1^\omega \times DPR_2^\omega \times \ldots$. Similarly to E defined in Section 3.2 for truly random channels, define $EPR_n = \{(\omega, \mathcal{S}) \mid |\omega| = n, s_{i,j} \in DPR_i^\omega\}$.

Because $\overrightarrow{DPR}^\omega$ has the requisite min-entropy, it is valid to expect proper performance of the stegoencoder on it; because it is pseudorandom, an analog of Lemma 2 will still hold; and because it has efficient membership testing given a short seed, the adversary will be able to see if an output of the stegoencoder is not from it.

We are now ready to formally state the claim about the properties of $\overrightarrow{DPR}$. For this claim, and for the rest of the proof, we assume existence of a family of pseudorandom functions $\mathcal{F}$ with insecurity $\mathbf{InSec}_{\mathcal{F}(n)}^{\mathrm{PRF}}(t, d, q)$ (recall $\mathbf{InSec}$ 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 note that for us d always will be at most description size of the stegosystem plus some constant c_1, and that $q \leq t$. We will then write $\iota_{PRF}(n, t)$ instead of $\mathbf{InSec}_{\mathcal{F}(n)}^{\mathrm{PRF}}(t, d, q)$.

Claim 1. *There is a polynomial p and a family of channels $\overrightarrow{DPR}^\omega$, indexed by a string ω of length n (as well as values H and S), such that for any positive integers n, i and $H \leq S$, channel $\overrightarrow{DPR}$ has the following properties:*

- *is a flat h-channel for $h = \log H$ on the alphabet $\{0, \ldots, S-1\}$;*
- *allows for sampling and membership testing in time polynomial in $n, \log S$ and $\log i$ given ω, i, H and S as inputs;*

- *is pseudorandom in the following sense: for any H, S, and any oracle machine (distinguisher) A with running time $\tau \geq \log S$,*

$$\left| \Pr_{(\vec{D},\mathcal{S})\leftarrow E}[A^{\mathcal{S},Memb(\vec{D})}() = 1] - \Pr_{(\omega,\mathcal{S})\leftarrow EPR_n}[A^{\mathcal{S},Memb(\omega)}() = 1] \right| < \iota_{PRF}(n, p(\tau, n)) + \tau 2^{-n} \, ,$$

where $Memb(\vec{D})$ and $Memb(\omega)$ denote membership testing oracles for $\vec{D}$ and $\overrightarrow{DPR}^{\omega}$, respectively.

The claim follows from the results of [8] with minor modifications, as presented above. We present no proof here.

Note that the second argument to ι_{PRF} depends on S only to the extent τ does; this is important because even for large alphabets and high-entropy channels, because we want to keep the second argument to ι_{PRF} as low a possible so that ι_{PRF} is as low as possible.

Stegosystems Running with *DPR* Almost Always Output Query Answers Having built pseudorandom channels, we now state the analog of Lemma 2 that works for stegosystems secure only against bounded adversaries. Fix some H and S. Let A be the same as in Lemma 2, but given access to $\overrightarrow{DPR}^{\omega}$ instead of $\vec{D}$, and let $t = t_1 \dots t_l$ be its output and Q_i be the set of responses A received to its queries of the ith channel DPR_i. Analogously to Nq and Ins, define the following two families of events, indexed by n, the security parameter for the PRF.

- nonqueried, pseudorandom version: $NqPR_n = \{(\omega, \mathcal{S}) \in EPR_n \mid (\exists i) t_i \notin Q_i\}$
- in support, pseudorandom version: $InsPR_n = \{(\omega, \mathcal{S}) \in EPR_n \mid t \in \overrightarrow{DPR}^{\omega}\}$

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

Lemma 3. *There exists a polynomial p_1 such that for any A running in time $\tau \geq \log S$, if $\Pr[\overline{InsPR_n}] < \epsilon(n)$, then*

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

Proof. Let Ins and Nq be the same as in Lemma 2. Let A' be a machine that is given an oracle which tests membership in the channel. Let A' run A to get t and output 1 if and only if the membership oracle says that t is in the channel. Applying Claim 1 to A', we have that for some polynomial p' (namely, the polynomial $p(\tau + t_{A'}(\tau), n)$, where $t_{A'}$ is the extra time that A' needs to run after A is finished),

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

Therefore $\Pr[\overline{Ins}] < \epsilon(n) + \iota_{PRF}(n, p(\tau + p'(\tau, n))) + \tau 2^{-n}$. It now follows, by the same derivation as for Equation (1) in the proof of Theorem 1, that

$$\Pr[Nq] < \frac{\epsilon(n) + \iota_{PRF}(n, p'(\tau, n)) + \tau 2^{-n}}{1 - H/S} \, .$$

Let A'' be a machine that runs A and outputs 1 if and only if A outputs something it did not receive as a query response. Applying Claim 1 to A'', we get that for some polynomial p'' (namely, the polynomial $p(\tau + t_{A''}(\tau), n)$, where $t_{A''}$ is the extra time that A'' needs to run in addition to A), we get $|\Pr[\mathit{NqPR}_n] - \Pr[\mathit{Nq}]| < \iota_{PRF}(n, p''(\tau, n)) + \tau 2^{-n}$. Therefore,

$$\Pr[\mathit{NqPR}_n] < \frac{\epsilon(n) + \iota_{PRF}(n, p'(\tau, n))}{1 - H/S} + \iota_{PRF}(n, p''(\tau, n)) + (1 + R)\tau 2^{-n} .$$

Now let $p_1 \geq \max(p', p'')$. □

Completing the Proof. We are now ready to prove Theorem 2. We define the same events as in the proof of Theorem 1, except as subsets of $\mathit{EPR}_n \times \{0,1\}^* \times \{0,1\}^{lw} \times \{0,1\}^*$ rather than $E \times \{0,1\}^* \times \{0,1\}^{lw} \times \{0,1\}^*$ (we use the suffix PR to emphasize that they are for the pseudorandom channel): FewPR_n, CorrPR_n, LowPR_n denote, respectively, that SE made at most N queries, that SD correctly decoded the hiddentext, and that the hiddentext has a low-weight encoding.

Just like in the proof of 1, it holds that $\Pr[\mathit{FewPR}_n] \leq \Pr[\mathit{LowPR}_n] + \Pr[\mathit{NqPR}_n] + \Pr[\overline{\mathit{CorrPR}_n}]$, and that $\Pr[\overline{\mathit{CorrPR}_n}] < \rho$ and $\Pr[\mathit{LowPR}_n] < (Ne/l2^w)^l$. It is left to argue a bound on $\Pr[\mathit{NqPR}_n]$.

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 $\overrightarrow{\overline{DPR}}^{\omega}$. It can be implemented to run in $p_2(n, \log S, l)$ steps for some polynomial p_2, and has description size $n + c_2$ for some constant c_2. Hence its probability of detecting a stegoencoder output that is not in $\overrightarrow{\overline{DPR}}^{\omega}$ cannot be more than the insecurity $\epsilon = \mathbf{InSec}^{\mathrm{SS}}_{ST(\kappa), \overrightarrow{\overline{DPR}}^{\omega}}(p_2(n, \log S, l), n + c_2, 1, lw)$. In other words, $\Pr[\overline{\mathit{InsPR}_n}] \leq \epsilon$, and, by Lemma 3, we get

$$\Pr[\mathit{NqPR}_n] \leq R\epsilon + (R + 1)(\iota_{PRF}(n, p_1(\tau, n)) + \tau 2^{-n}) .$$

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: specifally, Lemma 3 relies on Lemma 2. 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 [8] 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 [8]. 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[3] to that of Theorem 2. We see no such opportunity

[3] 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.

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

Our encoder is similar to the rejection-sampler-based encoder of [11] 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 [12, 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 $m = m_1 m_2 \ldots m_l$, 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 $s = s_1 s_2 \ldots s_l$
```

```
Procedure STF.$SD(K, w, s, ctr)$:
    Let $s = s_1 \ldots s_l$, 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 f becomes 1 after j iterations of the inner loop in STF.*SE* (i.e., 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) [17, 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 $2^{-h}(2^w-1) \le 1/2$ and using $\log(1-x) \ge -2x$ for $0 \le x \le 1/2$, we get that the capacity of the channel created by the encoder is at least $w + 2^{-h}(2^w-1)(-h-2) \ge w - (h+2)2^{-h+w}$. 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 [11]). 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*

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

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 to 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

```
$SE_2(K, w, k, m_1 \dots m_l, \mathcal{H})$:
$Q \leftarrow \emptyset$
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

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.

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,*

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

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 $G(s_{i,j}) = m_i$, no such $s_{i,j}$ has been drawn. Second, if the same value is returned twice by M before SE_2 finds a satisfactory $s_{i,j}$; in other words, if there has been a collision between two unsuccessful covertext documents.

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

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

Observe that the probability that SE_2 outputs a specific document s which is not "Fail" can be only 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',

$$\begin{aligned}
\Pr[E_3'] &\leq \frac{p}{R^2} \sum_{n_1=1}^{k} \sum_{n_2=1}^{k} \sum_{n_3=1}^{k} \left(\frac{R-1}{R}\right)^{n_1+n_2+n_3-2} (n_1+n_2+n_3-1) \\
&= \frac{p}{R^2} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(R\Pr[E_2']/p + n_1 \sum_{n_2=1}^{k} \left(\frac{R-1}{R}\right)^{n_2-1} \sum_{n_3=1}^{k} \left(\frac{R-1}{R}\right)^{n_3} \right) \\
&< \frac{p}{R^2} \sum_{n_1=1}^{k} \left(\frac{R-1}{R}\right)^{n_1-1} \left(R\Pr[E_2']/p + n_1 R(R-1)\right)
\end{aligned}$$

$$\begin{aligned} &< \frac{p}{R^2}\left(R^2\Pr[E_2']/p + R^3(R-1)\right) \\ &= p(3R-1)(R-1)\,. \end{aligned}$$

In general, for E_i' we have the recurrence,

$$\begin{aligned} \Pr[E_i'] &\le \frac{p}{R^{i-1}}\sum_{n_1=1}^{k}\left(\frac{R-1}{R}\right)^{n_1-1}\left(R^{i-2}\Pr[E_2']/p + n_1R^{i-2}(R-1)\right) \\ &< \Pr[E_{i-1}'] + pR(R-1)\,, \end{aligned}$$

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_i fails. This occurs because either: (1) there are k unsuccessful attempts to find $s_{i,j}$ such that $G(s_{i,j}) = m_i$; 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*

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

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.

The stegoencoder fails to encode properly when it cannot find $s_{i,j}$ such that $G(s_{i,j}) = m_i$ after k attempts. We will consider separately the case when G is simply unlikely to hit m_i, and when G is reasonably likely to hit m_i, but the samples from the channel are just unlucky for k times in a row.

To bound the probability of failure in the first case, fix some channel history $\mathcal{H}$ and w-bit message m, and consider the probability over G 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}). Formally, consider $\Pr_G[\Pr_{s \leftarrow D_{\mathcal{H}}}[G(s) = m] < c2^{-w}]$. 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 $\Pr_{s \leftarrow D_{\mathcal{H}}}[G(s) = m] = 1 - \sum_{i=1}^{n} X_i$. Note that the expected value, over G, of $\sum_{i=1}^{n} X_i$ is $1 - 2^{-w}$. Using Hoeffding's inequality (Theorem 2 of [10]), we obtain

$$\begin{aligned}
\Pr_G[1 - \sum_{i=1}^{n} X_i \le c2^{-w}] &\le \exp\left[-2(1-c)^2 2^{-2w} / \sum_{i=1}^{n} p_i^2\right] \\
&\le \exp\left[-2(1-c)^2 2^{-2w} / 2^{-h} / \sum_{i=1}^{n} p_i\right] \\
&= \exp\left[-2(1-c)^2 2^{h-2w}\right],
\end{aligned}$$

where the second to last step follows from $p_i \le 2^{-h}$ 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 messages $m \in \{0,1\}^w$, we get that the probability that G is skewed for at least one message is at most $2^w \exp\left[-2^{h-2w-1}\right]$.

To bound the probability of failure in the second case, assume $G(D_{\mathcal{H}})$ is not so skewed. Then the probability of failure is

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

The result follows from setting $c = 1/2$ and taking the union bound over l. □

Acknowledgements

We are grateful to Nick Hopper for clarifying related work and to anonymous referees for their helpful comments.

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. In Kilian [13], pages 210–226.

[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 International 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 Kilian [13], pages 227–244.

[6] Bert Fristedt and Lawrence Gray. *A Modern Approach to Probability Theory.* Birkhäuser, 1997.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] Joe Kilian, editor. *Second Theory of Cryptography Conference — TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

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

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

[16] 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.

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

[18] 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/`.

[19] 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 [11, 19, 1] (particularly public-key ones) use the following approach: they encrypt the hiddentext 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 resulting ciphertext.

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 [18], 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 $\sqrt{l}2^{(-h+w)/2}$.