Steganography with Imperfect Samplers

Anna Lysyanskaya
Brown University
anna@cs.brown.edu

Maria Meyerovich
Brown University
mira@cs.brown.edu

December 7, 2005

Abstract

The goal of steganography is to pass secret messages by disguising them as innocent-looking covertexts. Real world stegosystems are often broken because they make invalid assumptions about the system's ability to sample covertexts. We examine whether it is possible to weaken this assumption. By modeling the covertext distribution as a stateful Markov process, we create a sliding scale between real world and provably secure stegosystems. We also show that insufficient knowledge of past states can have catastrophic results.

Keywords Steganography, information hiding, digital signature, Markov process

1 Introduction

The goal of steganography is to pass secret messages by sending innocuous data. The sender may give the receiver *covertexts* that are distributed according to a *covertext distribution*. A covertext is made up of multiple *documents*. For example, a digital camera can define a covertext distribution of photographs, in which pixels, tiles, or even entire pictures can be considered documents. A *stegosystem* transforms a secret message, called a *hiddentext*, into a *stegotext* that looks like a covertext.

Real-world stegosystems are broken because they make invalid assumptions about the covertext distribution. Often, this is an assumption about an *adversary's lack of knowledge* about the distribution. For example, for a long time, modifying the least significant bits of pixels values in bitmaps was considered a good idea because these bits looked random. Then Moskowitz, Longdon and Chang [MLC01] showed that there is a strong correlation between the least significant bit and the most significant bit (see Figures 7-10 in their paper for an instructive example).

Provably secure steganography attacked the problem by quantifying the *stegosystem's need for knowledge*. Anderson and Petitcolas [AP98] observe that every covertext can be compressed to generate a hiddentext. Therefore, to hide a message, we "decompress" it into a stegotext. Le [Le03] and Le and Kurosawa [LK03] construct a provably secure compression based stegosystem that assumes both the sender and receiver know the covertext distribution exactly. Independently, Sallee [Sal03] implemented a compression-based stegosystem for JPEG images that lets the sender and receiver estimate the covertext distribution. Compression-based schemes need to know the exact probability of every possible covertext.

Cachin [Cac98] proposed using rejection-sampling to generate stegotexts that look like covertexts. A publicly known hash function assigns a bit value to documents. To send one bit, the stegosystem samples from the covertext distribution until it selects a document that evaluates to the message XOR K, where K is a session key both parties derive from their shared secret key. Sending multiple bits requires stringing several documents together. Cachin's scheme is secure if the hash function is unbiased. Because the stegosystem only needs to be able to sample from the covertext distribution, it is known as a *black-box* stegosystem. This paper examines the nature of the black-box required for steganography.

Hopper, Langford and von Ahn [HLvA02] improve on Cachin's results. They give the first rigorous definition of steganographic security by putting it in terms of computational indistinguishability from the

covertext distribution. Their stegosystem uses Cachin's rejection-sampling technique, but generalizes it to be applicable to any distribution, assuming it (1) has sufficient entropy and (2) can be sampled perfectly based on prior history. Reyzin and Russell [RR03] improve the robustness and efficiency of the Hopper et al. scheme. Von Ahn and Hopper [vAH04] create a public-key provably secure stegosystem and Backes and Cachin [BC05] and Hopper [Hop05] consider chosen covertext attacks. Despite these improvements, the two assumptions necessary for provably secure steganography remain in the literature. The entropy assumption appears inherent to the problem. We address the possibility of weakening the sampling assumption.

Some prior work focuses on the performance measures of black-box stegosystems. In particular, there is the *rate* of a stegosystem, which measures how many bits of the message you can pack per document transmitted. There is also the *query complexity per document* which measures how many times you need to query the sampler in order to create a document of the stegotext. Notably, Dedic et al. [DIRR05] showed that if the rate is w, then the query complexity per document is 2^w. We do not worry about query complexity, but rather about the very nature of the sampler at the disposal of a stegosystem, so the underlying question is very different.

Black-box stegosystems [Cac98, HLvA02, RR03, vAH04, BC05, Hop05] assume that they have access to an *adaptive* sampler. The sampler must be able to take an arbitrary history of documents as input and output a document distributed according to the covertext distribution conditioned on the prior history. For example, if our covertext distribution consists of images of teddy-bears, and each document is an 8×8 pixel tile, then the sampler's input is the first $k-1$ tiles of the image (say, the ears of the teddy bear), and the output is the k^{th} tile of the image (say, the nose). The stegosystem needs to be able to query the sampler multiple times on the same input: it continues to sample until it gets a document that corresponds to the message it wants to hide. The sampler must output many noses that correspond to the same set of ears.

Sampling teddy-bear noses based on teddy-bear ears is an absurd example. We use it because in the real world there are no known covertext distributions that can be sampled based on history. Our work examines whether accurate adaptive sampling is really neccessary. We come to the somewhat unsurprising conclusion that a stegosystem must assume that the sampler it uses is accurate. Our chief contribution is to examine what it really means to have a bad sampler.

There are many ways to characterize the abilities of a sampler. It can be contextual: given documents $d_i, \ldots, d_{j-1}, d_{j+1}, \ldots, d_k$, it produces possible values for d_j. A special case of a context sampler is a history based sampler: given $d_i, \ldots, d_{j-1}$, it produces possible values for d_j. Since history-based samplers are sufficient for secure steganography, we limit our examination to those. Past experience has shown that stegosystems are broken when there is a statistical correlation between documents of the covertext distribution. For example, the least-significant and most-significant bits in a bitmap are correlated, which leads to Moskowitz et al's [MLC01] attack. Therefore, a history-based sampler might make a mistake *when it does not consider some of the history*. This means we can characterize a history-based sampler by the length of history it considers. We call a sampler that considers only some of the history a *semi-adaptive* sampler, while one that ignores the history entirely is called *non-adaptive*.

Semi-adaptive samplers lead us naturally to consider Markov processes. Suppose the actual covertext distribution is D. The distribution D' from which a semi-adaptive sampler draws is a Markov process. Since a stegosystem approximates the distribution it samples, security requires that D and D' are sufficiently close. We introduce the concept of an *α-memoryless distribution*, a distribution that is computationally indistinguishable from some Markov process of order α. We design the definition of α-memorylessness so that it is necessary and sufficient for secure black-box steganography with semi-adaptive sampling.

We have three results:

1. We analyze what happens to the von Ahn and Hopper public key stegosystem [vAH04] when the sampler only considers the last α documents of the history. We calculate how inaccuracy in the sampler translates into insecurity in the stegosystem. Our results show that assuming the covertext distribution is α-memoryless is neccessary and sufficient for maintaining security.

2. We analyze the security of non-adaptive black-box stegosystems. Independently,[1] Petrowski et al. [PKSM]

[1] We presented preliminary results of this work in August 2004 [LM04].

implemented a non-adaptive stegosystem for JPEG images, giving empirical evidence that memoryless distributions exist and can be used for secure steganography.

3. We construct a pathological α-memoryless high-entropy distribution for which black-box steganography is infeasible if the stegosystem's sampler considers only the last $\alpha-1$ documents of the history (under the discrete logarithm assumption). An efficient adversary can detect any attempt at covert communication with overwhelming probability.

Organization: Section 2 presents notation and definitions. Section 3 analyzes the von Ahn and Hopper stegosystem [vAH04] in the context of semi-adaptive sampling. Section 4 examines non-adaptive stegosystems. Section 5 constructs a pathological covertext distribution for which black-box steganography is infeasible. Section 6 concludes.

2 Notation

We call a function $\nu\colon \mathbb{N} \to (0,1)$ *negligible* if for all $c > 0$ and for all sufficiently large k, $\nu(k) < 1/k^c$.

The hiddentext will always be in $\{0,1\}^*$. A covertext is composed of a sequence of documents. Each document comes from the alphabet $\mathbb{A}$; $|\mathbb{A}|$ may be exponential. We denote concatenation with the $\circ$ operator; a string s can be parsed to $s = s_1 \circ s_2 \circ ... \circ s_n$, where $|s| = n$. The symbol λ denotes the empty string.

We say that a function $f : \mathbb{A} \to \{0,1\}$ is ϵ-*biased* with respect to distribution D if $|Pr[d \leftarrow D : f(d) = 0] - 1/2| < \epsilon$. A $\epsilon(k)$-biased function is called an *unbiased* function if ϵ is a negligible function.[2] A covertext distribution that has sufficient minimum entropy for steganography is called *always informative* (see Hopper et al [HLvA02] for details).

We write $x \leftarrow D\langle h, n\rangle$ to denote sampling n documents from D conditioned on the prior history h; $D\langle h, n\rangle$ defines a distribution over $\mathbb{A}^n$. A semi-adaptive sampler $D^\alpha\langle h\rangle$ samples one document from the distribution D conditioned only on the last α documents of h. For ease of exposition, we introduce the notation $D^\alpha\langle h, n\rangle$ to mean generating an n-document string by calling the semi-adaptive samper n times, each time appending the result to h. This is syntactic sugar; anything that can be done with $D^\alpha\langle h, n\rangle$ can be done using only $D^\alpha\langle h\rangle$.

An α-memoryless distribution is indistinguishable from a Markov process of order α. (A sequence of random variables $X_1, \ldots, X_n$ such that for $\alpha < i \leq n$, the conditional distribution $\{X_i \mid X_{i-\alpha}, \ldots, X_{i-1}\}$ is identical to the conditional distribution $\{X_i \mid X_1, \ldots, X_{i-1}\}$.) Since we require computational indistinguishability, we parameterize everything by k (e.g. D_k, a family of distributions).

Definition 2.1 (α-Memoryless). *Let D_k be a family of distributions indexed by a public parameter k and let D_k^α be the best Markov model of order α that approximates D_k. We define the advantage of an adversary A against the Markov model as:*

$$\mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(A,k) = |Pr[h \leftarrow D_k\langle\lambda, n(k)-1\rangle; x \leftarrow D_k^\alpha\langle h, 1\rangle : A(h \circ x) = 1] - Pr[x \leftarrow D_k\langle\lambda, n(k)\rangle : A(x) = 1]|$$

We let $\mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t,n,k) = \max_{A\in\mathcal{A}(t,n,k)} \mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(A,k)$, where $\mathcal{A}(t,n,k)$ is the set of all adversaries that run in time $t(k)$ and get a sample $n(k)$ documents long. We say that D_k is α-memoryless if $\mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t,n,k) \leq \nu(k)$ for some negligible function ν. D_k is strictly α-memoryless if $\mathbf{InSec}^{\mathsf{mem}}_{D,\beta}(t,n,k)$ is non-negligible for all $\beta < \alpha$.

Remark This property is necessary and sufficient for steganography with semi-adaptive sampling.

[2] The function f is typically chosen after we fix the distribution (and the security parameter). A universal hash function is often used in practice.

2.1 Standard Cryptographic Notions

We define $\mathbf{InSec}^{\mathsf{dist}}_{X,Y}(t,n,k)$ as the maximum probability that an adversary can distinguish distribution X_k from Y_k if it runs in time $t(k)$ and gets a $n(k)$ document long sample.

Definition 2.2 (Indistinguishability). *Let $\{X_i\}$ and $\{Y_i\}$ be two families of distributions indexed by a public parameter k. We say the advantage of an adversary A trying to distinguish X from Y is:*

$$\mathbf{Adv}^{\mathsf{dist}}_{X,Y}(A,k) = |Pr[x \leftarrow X_k : A(x) = 1] - Pr[y \leftarrow Y_k : A(y) = 1]|$$

$\mathbf{InSec}^{\mathsf{dist}}_{X,Y}(t,n,k) = \max_{A \in \mathcal{A}(t,n,k)} \mathbf{Adv}^{\mathsf{dist}}_{X,Y}(A,k)$ *where $\mathcal{A}(t,n,k)$ is the set of all adversaries that run in $t(k)$ time and get a challenge string $n(k)$ documents long. We say that X and Y are indistinguishable ($X \approx Y$) if $\mathbf{InSec}^{\mathsf{dist}}_{X,Y}(t,n,k) \leq \nu(k)$ for some negligible function ν.*

Steganography requires an IND$-CPA cryptosystem whose ciphertext is indistinguishable from random. $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t,q,n,k)$ is the insecurity of cryptosystem $\mathcal{E}$ against a chosen plaintext attack by an adversary that runs in $t(k)$ time, makes $q(k)$ queries and gets responses totaling $n(k)$ bits.

Definition 2.3 (IND$-CPA). *Let $\mathcal{E} = (G,E,D)$ be a public-key cryptosystem; G generates the public/secret-key pair, E is the encryption function, and D is the decryption function. Let R be the uniform distribution over $\{0,1\}^*$ such that $\forall m : |R(m)| = |E_{PK}(m)|$. The advantage of an adversary A against $\mathcal{E}$ in a chosen plaintext attack (CPA) is:*

$$\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(A,k) = \left| Pr[PK \leftarrow G(1^k) : A^{E(PK,\cdot)}(PK) = 1] - Pr[A^{R(\cdot)}(PK) = 1] \right|$$

We let $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t,q,n,k) = \max_{A \in \mathcal{A}(t,q,l,k)} \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(A,k)$, where $\mathcal{A}(t,q,l,k)$ is the set of all adversaries that run in $t(k)$ time and issue $q(k)$ queries, getting a response totaling $n(k)$ bits. We say that $\mathcal{E}$ is indistinguishable from random under an adaptive chosen plaintext attack (IND$-CPA) if $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t,q,n,k) \leq \nu(k)$ for some negligible function ν.

The pathological covertext distribution we construct in Section 5.1 requires a secure stateless signature scheme.

Definition 2.4 (Stateless Signature Scheme). *A stateless signature scheme $\Sigma = (G,\sigma,V)$ is a triple of polynomial time algorithms such that:*

1. *$G(1^k)$ is a probabilistic algorithm that generates a k-bit signing key SK and k-bit verification key VK.*
2. *$\sigma : \{0,1\}^k \times \mathcal{M}_k \rightarrow \{0,1\}^{p(k)}$ is a probabilitic algorithm that on input $\sigma(SK,m)$ outputs a $p(k)$ bit signature on message m using the signing key SK.*
3. *$V : \{0,1\}^k \times \mathcal{M}_k \times \{0,1\}^{p(k)} \rightarrow \{0,1\}$ is the standard verification function that takes the verification key VK, a message m and $p(k)$-bit signature as input.*

Definition 2.5 (Secure Signature Scheme). *The advantage of adversary A against the signature scheme $\Sigma = (G,\sigma,V)$ in an adaptive chosen message attack is:*

$$\mathbf{Adv}^{\mathsf{sig}}_{\Sigma}(A,k) = Pr[(SK,VK) \leftarrow G(1^k); (Q,m,s) \leftarrow A^{\sigma(SK,\cdot)}(VK) : V(VK,s,m) = 1 \text{ and } m \notin Q]$$

(The adversary A must honestly record all of its queries to $\sigma(SK,\cdot)$ on the query tape Q.)

We let $\mathbf{InSec}^{\mathsf{sig}}_{\Sigma}(t,q,k) = \max_{A \in \mathcal{A}(t,q,k)} \mathbf{Adv}^{\mathsf{sig}}_{\Sigma}(A,k)$, where $\mathcal{A}(t,q,k)$ is the set of all adversaries that run in time $t(k)$ and make $q(k)$ queries to $\sigma(SK,\cdot)$. We say that Σ is a secure signature scheme if $\mathbf{InSec}^{\mathsf{sig}}_{\Sigma}(t,q,k) \leq \nu(k)$ for some negligible function ν.

Goldreich [Gol04] shows that *stateless* signature schemes exist if one-way functions exist. It is also known that the discrete logarithm assumption implies one-way functions. Therefore, the discrete logarithm assumption also implies the existence of stateless signature schemes. We let $\mathbf{DL}(t,k)$ be the maximum probability that any algorithm running in time $t(k)$ can solve the discrete logarithm problem:

Definition 2.6 (Discrete Logarithm Assumption). *Let G be a group of order p, where p is a k-bit prime, and let g be a generator of G. The advantage of an adversary A in computing the discrete log is:*

$$\mathbf{Adv}^{\mathsf{dl}}_G(A,k) = Pr[x \leftarrow \mathbb{Z}_p; y \leftarrow A(g^x, G, g, p) : g^y = g^x]$$

Let $\mathbf{DL}(t,k) = \max_{A \in \mathcal{A}(t,k)} \mathbf{Adv}^{\mathsf{dl}}_G(A,k)$, where $\mathcal{A}(t,k)$ is the set of all adversaries that run in time $t(k)$. The discrete logarithm assumption states that for certain G, $\mathbf{DL}(t,k) \leq \nu(k)$ for some negligible function ν.

2.2 Steganographic Notions

The following definitions are either standard or come from von Ahn and Hopper [vAH04]. We assume that all adversaries are probabilistic polynomial-time Turing machines. However, the distributions we work with are arbitrary and may act as arbitrarily powerful adversaries.

The standard specification [vAH04] of a public-key stegosystem is:

Definition 2.7 (Public Key Stegosystem). *A public key stegosystem is the triple $\mathcal{S} = (SG, SE, SD)$. $SG(1^k)$ generates a key-pair (SK, PK). $SE(PK, m)$ takes the public key PK and a message $m \in \{0,1\}^*$, and returns some stegotext s. $SD(SK, s)$ takes the secret key SK and stegotext s and returns a hiddentext m. For all $m \in \{0,1\}^*$, the probability that $SD(SK, SE(PK, m))$ fails to recover m should be negligible.*

Von Ahn and Hopper [vAH04] define the security of a public-key stegosystem against a chosen hiddentext attack. An adversary A queries an oracle with hiddentexts. The oracle responds either with stegotexts generated by $SE(PK, \cdot)$ or with covertexts of the appropriate length, generated by $D^*(\cdot)$. A should not be able to distinguish the two cases.

Definition 2.8 (SS-CHA). *The advantage of an adversary A against a public-key stegosystem $S = (SG, SE, SD)$ in a chosen hiddentext attack (CHA) is:*

$$\mathbf{Adv}^{\mathsf{cha}}_{S,D}(A,k) = \left| Pr[PK \leftarrow SG(1^k) : A_k^{SE^D(PK,\cdot),D} = 1] - Pr[A_k^{D^*(\cdot),D} = 1] \right|$$

We let $\mathbf{InSec}^{\mathsf{cha}}_{S,D}(t,q,n,k) = \max_{A \in \mathcal{A}(t,q,n,k)} \mathbf{Adv}^{\mathsf{cha}}_{S,D}(A,k)$ where $\mathcal{A}(t,q,n,k)$ is the set of all adversaries that run in $t(k)$ time, make $q(k)$ queries and get responses totaling $n(k)$ bits. A stegosystem is considered secure against a chosen hiddentext attack (SS-CHA) if $\mathbf{InSec}^{\mathsf{cha}}_{S,D}(t,q,n,k) \leq \nu(k)$ for some negligible function ν.

Remark We restrict the usual definition of security. Typically, the adversary is allowed to query the stegosystem with any history and message. In our model, we assume that an adaptive sampler does not exist. A stegosystem that is secure against such an attack is an adaptive sampler (see Hopper [Hop04] Section 3.3.2). We force the adversary to always query the stegosystem with history λ (the empty string).

3 Semi-adaptive stegosystem

In this section we examine what happens to the von Ahn and Hopper [vAH04] public-key stegosystem when we replace the adaptive sampling oracle with a semi-adaptive one. We show that if the oracle samples based on the last α documents of the history, then an α-memoryless distribution is necessary and sufficent for maintaining security.

3.1 The vAH04 Stegosystem with Semi-adaptive Sampling

The von Ahn and Hopper stegosystem [vAH04] (Construction 2 in their paper) is a public-key provably secure stegosystem. It uses an IND$-CPA public-key cryptosystem $\mathcal{E} = (G, E_{PK}, D_{SK})$ and a publicly known function $f : \Sigma \to \{0,1\}$ that is ϵ-biased with respect to the covertext distribution D_k. The encoder first encrypts the message using E_{PK}. Next, for each bit b of ciphertext, the encoder samples the covertext

distribution until it gets a document d such that $f(d) = b$. The encoder appends all of the resulting documents together to form the stegotext. The decoder extracts the ciphertext by evaluating f on every document of the stegotext and then decrypts the ciphertext. For the reader's convenience, we reprint the von Ahn and Hopper stegosystem using our notation; the encoder is defined in Algorithm 3.1 and the decoder is in Algorithm 3.2.

Algorithm 3.1: Encode

Input: Public key PK, message m, number of times to sample T

`step 1: Encrypt message`

$c \leftarrow E_{PK}(m)$;

`step 2: Stegocode ciphertext`

parse c as $c_1 \circ c_2 \circ ... \circ c_n$;
$h \leftarrow \varepsilon$;
for $j \leftarrow 1$ ***to*** n **do**
  $i \leftarrow 1$;
  repeat
    $s_j \leftarrow D_k^{\alpha}\langle h, 1\rangle$, increment i ;
  until $f(s_j) = c_j$ ***or*** $i > T$;
  $h \leftarrow h \circ s_j$;
end
$s \leftarrow s_1 \circ s_2 \circ ... \circ s_n$;
return s ;

Algorithm 3.2: Decode

Input: Secret key SK, stegotext s

`step 1: Extract ciphertext`

$c \leftarrow f(s_1) \circ f(s_2) \circ ... \circ f(s_n)$;

`step 2: Decrypt message`

$m \leftarrow D_{SK}(c)$;
return m

For the remainder of Section 3, we will refer to the von Ahn and Hopper stegosystem as $\mathcal{S} = (SG, SE, SD)$ and assume that D_k is the covertext distribution. We define a length function $\mathcal{L} : \mathbb{Z} \to \mathbb{Z}$ that calculates the length of a cipherphertext for a message m: $\mathcal{L}(|m|) = |E_{PK}(m)|$. Von Ahn and Hopper [vAH04] prove that their stegosystem is secure:

Theorem 3.1 ([vAH04]). *If D_k is an always informative distribution and f is ϵ-biased on D_k, then:*

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{S},D_k}(t, q, n, k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kn), q, n, k) + \mathcal{L}(n)\epsilon$$

Remark What Theorem 3.1 really states is that the output of $\mathcal{S}$ is indistinguishable from the distribution represented by its covertext oracle.

$\mathcal{S}$ uses a perfect sampler. We now consider the stegosystem $\mathcal{T} = (TG, TE, TD)$[3] that functions identically to $\mathcal{S}$, except that its only access to D_k is via D_k^{α}, an oracle that only considers the last α documents of the history. The main result of this section is the proof that $\mathcal{T}$ is correct and that $\mathcal{T}$ is secure if and only if D_k is α-memoryless.

[3] As a mneumonic device, think of $\mathcal{S}$ as the stegosystem with a Standard sampler and $\mathcal{T}$ as having a sampler that considers only the Tail of the history.

3.2 Analysis of $\mathcal{T}$

Lemma 3.2. *Assume that D_k is an always informative α-memoryless distribution and f is an ϵ-biased function on D_k. For all hiddentexts $m \in \{0,1\}^*$, the probability that $\mathcal{T}$ fails to encode m is negligible:*

$$Pr[(PK, SK) \leftarrow TG(1^k); s \leftarrow TE(PK, m); m' \leftarrow TD(SK, s) : m' \neq m] \\ \leq \mathcal{L}(|m|)(1/2 + \epsilon + \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(O(1), \mathcal{L}(|m|), k))^k$$

Proof. Suppose that f is β-biased on D^α_k for some value of β. The probability that the stegocoder fails to encode a single bit is at most $(1/2+\beta)^k$. Therefore, the probability that the stegocoder fails to correctly to encode the entire ciphertext is at most $\mathcal{L}(|m|)(1/2+\beta)^k$.

We calculate $|\epsilon - \beta|$ by creating an adversary A to distinguish D_k from D^α_k. A gets a challenge string $h \circ x$ of length at most $\mathcal{L}(|m|)$ where $h \leftarrow D_k\langle\lambda, n(k)-1\rangle$ and x is either generated by $D_k\langle h, 1\rangle$ or $D^\alpha_k\langle h, 1\rangle$. A calculates $f(x)$ and outputs the result. A's advantage is $|\epsilon - \beta| \leq \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(O(1), \mathcal{L}(|m|), k)$. We substitute for β to get the final result. □

Remark We note that $\mathcal{T}$ may still be correct if D_k is not α-memoryless (as long as f is unbiased). Also, decryption errors can be dealt with in a straightforward manner; details are omitted.

Theorem 3.3. *If D_k is an always informative α-memoryless distribution and f is ϵ-biased, then $\mathcal{T}$ is SS-CHA secure:*

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{T},D}(t,q,n,k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kn), q, n, k) + n\mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t + O(n), n, k) + \mathcal{L}(n)\epsilon$$

Proof. Let D_k be an always informative α-memoryless distribution and f be ϵ-biased. By construction of $\mathcal{T}$, we have:

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{T},D_k}(t,q,n,k) = \mathbf{InSec}^{\mathsf{cha}}_{\mathcal{S},D^\alpha_k}(t,q,n,k) + \mathbf{InSec}^{\mathsf{dist}}_{D_k,D^\alpha_k}(t,n,k)$$

We know from Theorem 3.1 that:

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{S},D^\alpha_k}(t,q,n,k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kn), q, n, k) + \mathcal{L}(n)\epsilon$$

To finish calculating the insecurity of $\mathcal{T}$ we need to determine the advantage of an adversary distinguishing D_k from D^α_k. We create a series of hybrid distributions $H_0, H_1, \ldots, H_n$, where H_i outputs i times from D_k and $n-i$ times from D^α_k. H_i differs from H_{i+1} only in position $i+1$. Suppose an adversary A can distinguish H_i from H_{i+1} based on a single sample. In that case we can create an adversary B to attack D^α_k. B gets an $i+1$ document long challenge h. B calls $h' \leftarrow D^\alpha_k\langle h, n-i-1\rangle$, then passes $h \circ h'$ to A. B outputs the same answer as A. Since B transforms samples from D^α_k into samples from H_i and samples from D_k into samples from H_{i+1}, $\mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(B,k) = \mathbf{Adv}^{\mathsf{dist}}_{H_i,H_{i+1}}(A,k)$. Suppose A runs in time t and gets samples of length n. B gets a sample of length i and uses $t + O(n-i-1)$ time to generate h' and run A on $h \circ h'$. As a result,

$$\mathbf{InSec}^{\mathsf{dist}}_{H_i,H_{i+1}}(t,n,k) \leq \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t + O(n-i-1), i, k)$$

By definition, $D_k = H_0$ and $D^\alpha_k = H_n$. Adding up the probabilities of distinguishing the hybrid distributions, we get that:

$$\mathbf{InSec}^{\mathsf{dist}}_{D_k,D^\alpha_k}(t,n,k) \leq \sum_{i=0}^{n-1} \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t + O(n-i-1), i, k) \\ \leq n\mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t + O(n), n, k)$$

We substitute the above expression for $\mathbf{InSec}^{\mathsf{dist}}_{D_k,D^\alpha_k}(t,n,k)$ and Theorem 3.1 to get the result in Theorem 3.3. □

Theorem 3.4. *Let D_k be an always informative distribution and f an ϵ biased function on D. If D_k is not α-memoryless then $\mathcal{T}$ is not a SS-CHA secure stegosystem. Specifically:*

$$\mathbf{InSec}^{\mathsf{cha}}_{\mathcal{T},D_k}(t+O(1),1,n,k) \geq \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t,n,k) - \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t+O(kn),1,n,k) - n\epsilon$$

Proof. Assume D_k is not α-memoryless. By definition, there exists an adversary A such that $\mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(A,k)$ is non-negligible. Let A run in time t and require a challenge sample of length n. We use A to create an adversary B that can tell whether it is querying an oracle representing $\mathcal{T}$ or D_k. B will ask its oracle for a single covertext of length n and pass the output to A. B will output whatever A outputs. B's advantage in distinguishing $\mathcal{T}$ from D_k is at least as much as A's advantage in distinguishing D^{α}_k from D_k minus the probability of distinguishing $\mathcal{T}$ from D^{α}_k:

$$\mathbf{Adv}^{\mathsf{cha}}_{\mathcal{T},D_k}(B,k) \geq \mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(A,k) - \mathbf{InSec}^{\mathsf{cha}}_{\mathcal{T},D^{\alpha}_k}(t,1,n,k)$$

Using Theorem 3.1, we get:

$$\mathbf{Adv}^{\mathsf{cha}}_{\mathcal{T},D_k}(B,k) \geq \mathbf{Adv}^{\mathsf{mem}}_{D,\alpha}(A,k) - \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t+O(kn),1,n,k) - n\epsilon$$

B runs in time $t+O(1)$ and gets 1 challenge string of length n, therefore:

$$\mathbf{InSec}^{\mathsf{cha}}_{S,D_k}(t+O(1),1,n,k) \geq \mathbf{InSec}^{\mathsf{mem}}_{D,\alpha}(t,n,k) - \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t+O(kn),1,n,k) - n\epsilon$$

This means that if D_k is not α-memoryless, then there exists an adversary that can launch a successful SS-CHA attack on $\mathcal{T}$ with non-negligible probability. □

Remark The above proof would probably work for any black-box stegosystem. However, because it is unclear how to deal with a stegosystem that somehow uses outside information (or how to rule out this possibility), we limit our analysis to the stegosystem $\mathcal{T}$.

4 Non-Adaptive Stegosystems

In this section, we show how to apply public-key black-box steganography as proposed by von Ahn and Hopper [vAH04] to real world covertext distributions. (Independently, Petrowski et. al. [PKSM] implemented a similar system for JPEG images, but their work had no security analysis.) The key insight is that multiple digital photographs of a still scene are almost but not completely identical. We can break up each image into 8×8 pixel tiles. A cryptographic hash function assigns a value to each tile. The stegosystem choses the appropriate tiles to create a composite photo that encodes the secret message. The scheme assumes each 8×8 pixel tile is independent of its neighbors.

This stegosystem is equivalent to using D^0_k to sample D_k and assuming that the covertext distribution is 0-memoryless, as shown in Algorithm 4.1. Non-adaptive steganography can be applied to any digital image format, TCP timestamp intervals, etc.

The analysis of Algorithm 4.1 follows directly from Section 3. Correctness: The probability that the stegosystem fails to encode a hiddentext m is: $\mathcal{L}(|m|)(1/2+\epsilon+\mathbf{InSec}^{\mathsf{mem}}_{D,0}(O(1),\mathcal{L}(|m|),k))^k$. Security: Algorithm 4.1 is secure if and only if D is 0-memoryless: an independent, but not necessarily identically distributed, sequence of random variables.

5 Pathological Covertext Distribution

In this section, we construct a pathological strictly α-memoryless distribution and prove that no computationally bounded algorithm can use it to hide messages without access to D^{α}_k. The distribution will publish

Algorithm 4.1: Non-adaptive stegosystem

Input: Public key PK, message m, T covertexts $x^{(1)}, \ldots, x^{(T)}$ (each covertext $x^{(i)}$ is of length $|E_{PK}m|$)

`step 1: Encrypt message`

$c \leftarrow E_{PK}(m)$;

`step 2: Stegocode ciphertext`

parse c as $c_1 \circ c_2 \circ ... \circ c_n$;
for $j \leftarrow 1$ ***to*** n **do**
  $i \leftarrow 1$;
  repeat
    $s_j \leftarrow x_j^{(i)}$, increment i ;
  until $f(s_j) = c_j$ ***or*** $i > T$;
end
$s \leftarrow s_1 \circ s_2 \circ ... \circ s_n$;
return s ;

a verification key that can be used by anyone to check if a covertext is legitimate. The probability that steganography will be detected is $1 - \nu(k)$.

We give a stegosystem a list of covertexts generated by $D\langle\lambda, \cdot\rangle$ and access to $D^{\alpha-1}\langle\cdot, 1\rangle$, a semi-adaptive oracle with insufficient memory. For example, a stegosystem might store a database of photographs (this corresponds to $D\langle\lambda, \cdot\rangle$) and maintain an internal Markov model about pixel color distributions based on the 8 adjacent pixels (this corresponds to $D^{\alpha-1}\langle\cdot, 1\rangle$, where $\alpha - 1 = 8$). We show that any stegotext produced by a stegosystem is really just a quote of a covertext in its database.

5.1 The Distribution

Our goal is to devise a covertext distribution where (1) each document depends on only the α documents that came before it (so it is α-memoryless); (2) a stegosystem cannot by itself compute the ith document d_i in a legitimate covertext; finally (3) it is very unlikely that the output of $D^{\alpha-1}\langle h, 1\rangle$ is a valid continuation of the last α documents of h.

The first construction that comes to mind is to make each document be a concatenation of a random number r_i and a signature on the previous α random numbers: $\sigma_i = \sigma(r_{i-\alpha}, \ldots, r_i)$. This will meet requirements (1) and (2). There is a subtle problem with this as far as requirement (3) is concerned. Suppose we are given $\alpha - 1$ documents $r_{n-\alpha+1}\sigma_{n-\alpha+1}, \ldots, r_{n-1}\sigma_{n-1}$. The signatures $\sigma_{n-\alpha+1}, \ldots, \sigma_{n-1}$ can leak partial information about the value $r_{n-\alpha}$. As a result, $D^{\alpha-1}\langle\cdot, 1\rangle$, even though not explicitly given $d_{n-\alpha}$, may nevertheless calculate $r_{n-\alpha}$ and compute the correct signature $\sigma_n = \sigma(r_{n-\alpha}, \ldots, r_n)$.

In order to fix this problem, we need to construct a signature function σ for which the following property holds: We fix a sequence of $2\alpha - 1$ integers $r_1, \ldots, r_{2\alpha-1}$. Then the sequence of $\alpha - 1$ documents $r_{\alpha+1}\sigma_{\alpha+1}, \ldots, r_{2\alpha-1}\sigma_{2\alpha-1}$ should be information theoretically independent of r_α. This property ensures that $D^{\alpha-1}$ cannot learn r_α and so will be unable to compute the correct signature $\sigma_{2\alpha}$ based on the previous α documents of h, as required by (3) above.

Consider the following hash function $h : \mathbb{Z}_p^\alpha \to G$, where p is a k-bit prime and G is a group of order p. The hash function $h_{p,G,g_1,\ldots,g_{\alpha+1}}$ is parameterized by p, G and $\alpha + 1$ generators of G: $g_1, \ldots, g_{\alpha+1}$. (We will omit the subscript of h in the future). On input $(r_1, \ldots, r_{\alpha+1}) \in \mathbb{Z}_p^{\alpha+1}$ the hash function returns:

$$h(r_1, r_2, \ldots, r_{\alpha+1}) \doteq g_1^{r_1} \cdot g_2^{r_2} \cdot \cdots \cdot g_{\alpha+1}^{r_{\alpha+1}}$$

The hash function h has the information hiding property that we need because it reveals only a linear combination of its inputs (see the proof of Lemma 5.7 in Section 5.2).

We now formalize the above discussion. We show how to modify a secure stateless signature scheme to use h and prove the result is secure under the discrete logarithm assumption. Then we construct our pathological distribution.

Construction 5.1. *Let $\Sigma' = (G', \sigma', V')$ be a secure stateless signature scheme that takes messages in $\{0,1\}^{2k}$ and outputs signatures in $\{0,1\}^{p(k)}$. We use (G', σ', V') and the hash function h to construct a new stateless signature scheme $\Sigma = (G, \sigma, V)$. We let $G = G'$.*

The signature function $\sigma : \{0,1\}^k \times (\mathbb{Z}_p^*)^{\alpha+1} \to \{0,1\}^{p(k)}$:

$$\sigma(SK, r_1 \circ \cdots \circ r_{\alpha+1}) = \sigma'(SK, h(r_1, \ldots, r_{\alpha+1}))$$

The verification function $V : \{0,1\}^k \times (\mathbb{Z}_p^*)^{\alpha+1} \times \{0,1\}^{p(k)} \to \{0,1\}$:

$$V(VK, s, r_1 \circ \cdots \circ r_{\alpha+1}) = V'(VK, s, h(r_1, \ldots, r_{\alpha+1}))$$

We further define σ on input from $(\mathbb{Z}_p^*)^\beta$, where $\beta < \alpha+1$ as follows: $\sigma(r_1, \ldots, r_\beta) = \sigma'(h(0, \ldots, 0, r_1, \ldots, r_\beta))$. V extends in the obvious way.

Lemma 5.2. *$\Sigma = (G, \sigma, V)$ from Construction 5.1 is a secure signature scheme under the discrete logarithm assumption:*

$$\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t, q, k) \leq \mathbf{InSec}_{\Sigma'}^{\mathsf{sig}}(t + O(q), q, k) + \mathbf{DL}(t + O(q), k)$$

Proof. Suppose there exists an adversary A such that $\mathbf{Adv}_{\Sigma}^{\mathsf{sig}}(A, k)$ is non-negligible. Then we can construct an algorithm B that will either break the security of $\Sigma' = (G, \sigma', V')$ or calculate discreet logs. B will get the public verification key VK and invoke $A(VK)$. Whenever A queries $\sigma(\cdot)$ with $r_1 \circ \cdots \circ r_{\alpha+1}$ B will intercept the message. B will calculate $u = h(r_1, \ldots, r_{\alpha+1})$. Then B will query $\sigma'(\cdot)$ with the input $r_{\alpha+1} \circ u$ and send the response to A. Eventually, with probability $\mathbf{Adv}_{\Sigma}^{\mathsf{sig}}(A, k)$ A will output a new message $m_1 \circ \cdots \circ m_{\alpha+1}$ and a valid signature s. B will take the output of A and output the message $m_{\alpha+1} \circ h(m_1, \ldots, m_{\alpha+1})$ and the signature s. V will accept the output of B if V' accepts the output of A. Assuming that A succeeded in forging, we have two cases we need to examine:

1. B has not previously queried σ with the message $m_{\alpha+1} \circ h(m_1, \ldots, m_{\alpha+1})$. In this case, B has made a successful forgery.
2. A has made a previous query $r_1 \circ \cdots \circ r_{\alpha+1} = m_{\alpha+1} \circ h(m_1, \ldots, m_{\alpha+1}$ but $r_1 \circ \cdots \circ r_{\alpha+1} \neq m_1 \circ \cdots m_{\alpha+1}$. Then we can use well know techniques to calculate discrete logarithms.

As a result, $\mathbf{Adv}_{\Sigma}^{\mathsf{sig}}(A, k) \leq \mathbf{Adv}_{\Sigma'}^{\mathsf{sig}}(B, k) + \mathbf{DL}(t + O(q), k)$. So $\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t, q, k) \leq \mathbf{InSec}_{\Sigma'}^{\mathsf{sig}}(t + O(q), q, k) + \mathbf{DL}(t + O(q), k)$. □

We use the signature scheme from Construction 5.1 to construct a distribution D_{VK} over the alphabet $\{\mathbb{Z}_p^* \times \{0,1\}^{poly(k)}\}^*$, where p is a k bit prime and $poly(k)$ is the length of a signature in Σ. Each document consists of an element in $\mathbb{Z}_p^*$ and a signature on the previous $\alpha + 1$ elements.

Construction 5.3 (Pathological Distribution D_{VK}). *Let $\Sigma = (G, \sigma, V)$ be a secure stateless signature scheme from Construction 5.1. We use G to generate the keys (SK, VK) and index distribution D_{VK} via the public verification key. If d_i is the ith document, then $d_i = r_i \sigma(SK, r_{i-\alpha} \circ \cdots \circ r_i)$, where r_i is chosen randomly from $\mathbb{Z}_p$. The output of $D_{VK}\langle \lambda, n \rangle$ looks like:*

$$\begin{aligned} D_{VK}\langle \lambda, n \rangle \to & r_1 \sigma(SK, r_1) \\ & \circ r_2 \sigma(SK, r_1 \circ r_2) \circ \cdots \\ & \cdots \circ r_{\alpha+1} \sigma(SK, r_1 \circ r_2 \circ \cdots \circ r_{\alpha+1}) \circ \cdots \\ & \cdots \circ r_n \sigma(SK, r_{n-\alpha} \circ \cdots \circ r_n) \end{aligned}$$

We define $\sigma_n = \sigma(SK, r_{n-\alpha}, \ldots, r_n)$.

Definition 5.4 (Γ). *Suppose we query $D_{VK}\langle\lambda, n\rangle$ q times and record the result on tape Q. We define the probability that any one sequence $r_1, \ldots, r_d$ appears two or more times in Q as $\Gamma(d, n, q, k)$.*

Lemma 5.5. *$\Gamma(d, n, q, k)$ is a negligible function in k.*

Proof. We make a rough estimate of the value of $\Gamma(d, n, q, k)$. $|\mathbb{Z}_p^*| = p - 1 \geq 2^{k-1}$. Therefore, there are at least 2^{k-1} possibilities for each document. We choose the first one at random. The probability that the second document matches the first is at most $1/2^{k-1}$. The probability that the ith document matches any of the previous $i - 1$ documents is at most $(i - 1)/2^{k-1}$. Since we sample qn documents, the probability of a match is at most

$$\frac{1}{2^{k-1}} \sum_{i=1}^{qn-1} i = \frac{1}{2^{k-1}} \cdot \frac{qn}{2} = qn2^{-k}$$

This is a very rough estimate; the probability would be even lower if we took d into account. □

5.2 Pathology of the Distribution

We now show that any computationally bounded stegosystem for D_{VK} is guaranteed to be caught with overwhelming probability.

Theorem 5.6. *Let $\mathcal{S}$ be an arbitrary probabilistic polynomial time stegosystem for distribution D_{VK} that has a database of q_1 covertexts of length n generated by $D_{VK}\langle\lambda, \cdot\rangle$ and is allowed to make q_2 queries to $D_{VK}^{\alpha-1}\langle\cdot, 1\rangle$. Suppose it takes $\mathcal{S}$ time t to generate a stegotext of length $N > \alpha$. Then there exists an adversary that can distinguish $\mathcal{S}$ from D_{VK} with probability $1 - \nu(k)$, for a negligible function ν. The adversary uses only the verification key VK and $q_1 + 1$ samples from the oracle of length N each; it runs in time $O((t + N)(q_1 + 1))$.*

Remark The stegosystem needs to forge signatures if it wants to generate more than q_1 distinct stegotexts. All the adversary does is examine the $q_1 + 1$ samples it gets for duplicates and/or invalid signatures.

We will prove Theorem 5.6 in three steps. First we will construct an oracle $D*_{VK}^{\alpha-1}$ that is information theoretically indistinguishable from $D_{VK}^{\alpha-1}\langle\cdot, 1\rangle$. Then we will show that a stegosystem whose only resource is $D*_{VK}^{\alpha-1}$ cannot create stegotexts longer than α with more than negligible probability. Finally, we will augment the stegosystem by giving it access to $D_{VK}\langle\lambda, \cdot\rangle$ and prove Theorem 5.6 by showing that it still cannot generate new stegotexts.

Algorithm 5.1: $D *_{VK}^{\alpha-1} \langle\cdot, 1\rangle$ with oracle access to $\sigma(SK, \cdot)$

Input: history: $h = r_1\sigma_1, \ldots, \circ r_{n-1}\sigma_{n-1}$
`If the history is more than` $\alpha - 1$ `documents long,` $D*_{VK}^{\alpha-1}$ `randomly chooses` $\hat{r}_n$ `and` $\hat{r}_{n-\alpha}$ `and signs the result.`
if $n < \alpha$ **then return** $D_{VK}\langle h, 1\rangle$;
else
  $\hat{r}_n \leftarrow$ Random ;
  $\hat{r}_{n-\alpha} \leftarrow$ Random ;
  $\hat{u} \leftarrow h(\hat{r}_{n-\alpha}, r_{n-\alpha+1}, \ldots, r_{n-1}, \hat{r}_n)$;
  $\hat{\sigma}_n \leftarrow \sigma(\hat{u})$;
end
return $\hat{r}_n\hat{\sigma}_n$;
`We use` $\hat{x}$ `to signify that the value of` x `was assigned by` $D *_{VK}^{\alpha-1} \langle\cdot, 1\rangle$

Lemma 5.7. *Consider $D *_{VK}^{\alpha-1} \langle\cdot, 1\rangle$ (Algorithm 5.1). $D *_{VK}^{\alpha-1} \langle\cdot, 1\rangle = D_{VK}^{\alpha-1}\langle\cdot, 1\rangle$.*

Proof. If the input to $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ represents the full history, its output is, by definition of D_{VK}, identical to $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$. We consider the case that $r_{n-\alpha+1}\sigma_{n-\alpha-1} \circ \cdots \circ r_{n-1}\sigma_{n-1}$ does not represent the full history. $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ needs to generate $r_n\sigma_n$. By definition of D_{VK}, the value of r_n is independent of everything that came before, so $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ can generate a random $\hat{r}_n$. The value σ_n is more complicated; it depends on $r_{n-\alpha}, \ldots, r_n$. Of these $\alpha + 1$ values, only the last α are known. The oracle $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ needs to calculate $r_{n-\alpha}$.

Assume for the moment that $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ is computationally unbounded. We need to show that it can do no better than $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ at guessing $r_{n-\alpha}$. Suppose that $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ can extract from a signature σ_x the values r_x and $u_x = h(r_{x-\alpha}, \ldots, r_x)$. In addition, suppose $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ can solve the discrete logarithm problem for some generator g. Then for each g_i used to compute the hash function h, $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ can find an x_i such that $g^{x_i} = g_i$. From this, $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ can perform the following computation:

$$u_{n-1} = g_1^{r_{n-\alpha-1}} \cdot g_2^{r_{n-\alpha}} \cdot \cdots \cdot g_{\alpha+1}^{r_{n-1}} = g^{x_1 r_{n-\alpha-1}} \cdot g^{x_2 r_{n-\alpha}} \cdot \cdots \cdot g^{x_{\alpha+1} r_{n-1}}$$
$$\log_g u_{n-1} = x_1 r_{n-\alpha-1} + x_2 r_{n-\alpha} + \cdots + x_{\alpha+1} r_{n-1}$$

Since $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ knows x_i, $\log_g u_{n-1}$ and $r_{n-\alpha+1}, \ldots, r_{n-1}$, it can calculate a value y_1 such that $x_1 r_{n-\alpha-1} + x_2 r_{n-\alpha} = y_1$. Via similar manipulations, it can establish a series of linear equations:

$$\begin{aligned} x_1 r_{n-\alpha-1} + x_2 r_{n-\alpha} &= y_1 \\ x_1 r_{n-\alpha-2} + x_2 r_{n-\alpha-1} + x_3 r_{n-\alpha} &= y_2 \\ &\vdots \\ x_1 r_{n-2\alpha+1} + \cdots + x_\alpha r_{n-\alpha} &= y_{\alpha-1} \end{aligned}$$

$D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ now has a series of $\alpha - 1$ linear equations with α variables $(r_{n-2\alpha+1}, \ldots, r_{n-\alpha})$. Algebraically, every value for $r_{n-\alpha}$ is equally likely. The value of $r_{n-\alpha}$ is information theoretically independent of the view of $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$. This means that if the oracle $D *_{VK}^{\alpha-1} \langle \cdot, 1\rangle$ chooses a random value for $r_{n-\alpha}$, its output will be information theoretically indistinguishable from $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$. □

Lemma 5.8. *D_{VK} is strictly α-memoryless.*

Proof. By Construction 5.3, each document $r_n\sigma_n$ depends on exactly α documents that came before it. Therefore, D_{VK} is α-memoryless. We now show D_{VK} is strictly α-memoryless. We construct an adversary A that gets an $\alpha+1$ document covertext $d = r_1\sigma_1 \circ \cdots \circ r_{\alpha+1}\sigma_{\alpha+1}$. The first α documents of d were generated by $d_1 \leftarrow D_{VK}\langle \lambda, \alpha\rangle$. The last document was generated by calling either $D_{VK}\langle d_1, 1\rangle$ or $D *_{SK}^{\alpha-1} \langle d_1, 1\rangle$; A's goal is to destinguish between the two cases. A uses the verification key VK to verify the signature in the last document and outputs 1 if it is legitimate, 0 otherwise. If d was generated by $D_{VK}\langle \lambda, \alpha + 1\rangle$ then A outputs 1 with probability 1. If it was generated with the help of $D *_{SK}^{\alpha-1} \langle d_1, 1\rangle$, A outputs 1 with probability $\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t, q, l, \alpha + 1, k)$ where $t(k)$ is the running time of $D *_{SK}^{\alpha-1} \langle d_1, 1\rangle$ and $q(k)$ is the number of queries of total length $l(k)$ it makes. So $\mathbf{Adv}_{D_{VK},\alpha-1}^{\mathsf{mem}}(A, k) = 1 - \mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t, q, l, \alpha + 1, k)$, which is clearly non-negligible. Using information theory, we know that $\forall \beta < \alpha$, $D_{VK}^{\beta}\langle \cdot, 1\rangle$ cannot approximate D_{VK} better than $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$. This means that D_{VK} is strictly α-memoryless. □

Lemma 5.9. *Let $\mathcal{S}$ be any stegosystem that has oracle access to $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$, but with no direct access to D_{VK} - i.e. $\mathcal{S}$ does not know SK and has no oracle access to $\sigma(SK, \cdot)$. Suppose it takes $\mathcal{S}$ t time and q queries to $D_{VK}^{\alpha-1}\langle \cdot, 1\rangle$ to output a stegotext $s = r_1\sigma_1 \circ \cdots \circ r_n\sigma_n$ of length $n > \alpha$. Then there exists an efficient adversary that can distinguish $\mathcal{S}$ from D_{VK} with overwhelming probability using only one text sample of length α and running in time $O(t)$:*

$$\mathbf{InSec}_{\mathcal{S},D_{VK}}^{\mathsf{cha}}(t, 1, \alpha + 1, k) \geq 1 - \mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t + O(1), q, k) - \mathbf{DL}(t + O(q), k)$$

Furthermore, $\forall i > \alpha$, the probability that an arbitrary signature σ_i is valid is at most:

$$\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t + O(1), q, k) + \mathbf{DL}(t + O(q), k)$$

Proof. Assume we have a secure stegosystem $\mathcal{S}$ with no direct access to D_{VK}. We construct an adversary A that uses $\mathcal{S}$ to forge signatures or calculate discrete logs. A tells S to generate a single stegotext of any length $n > \alpha$. While $\mathcal{S}$ is working, A intercepts all of $\mathcal{S}$'s queries to $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$ and redirects them to $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$. Finally, $\mathcal{S}$ outputs a stegotext $s = r_1\sigma_1 \circ r_2\sigma_2 \circ \cdots \circ r_n\sigma_n$.

Choose any $i > \alpha$. We have three cases to consider:

1. If σ_i is not a valid signature on $r_{i-\alpha} \circ \cdots \circ r_i$ then the stegosystem is insecure. The probability that this happens is $\mathbf{InSec}_S^{\mathsf{cha}}(t + O(1), 1, n, k)$.

2. If σ_i is a valid signature on $r_{i-\alpha} \circ \cdots \circ r_i$ and it was not generated by $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$ then $\mathcal{S}$ violated the security of Σ. The probability that this happens is $\mathbf{InSec}_\Sigma^{\mathsf{sig}}(t + O(1), q, k)$.

3. If σ_i is a valid signature that was generated by $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$ then we use $\mathcal{S}$ and $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$ to calculate discrete logarithms. The idea is that while $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$ only needs to know $g_1^{r_{n-\alpha}}$ to generate signature σ_n, $\mathcal{S}$ needs to output $r_{n-\alpha}$ in the clear as part of the stegotext.

Algorithm 5.2: $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$ with oracle access to $\sigma(SK,\cdot)$

Input: history: $r_1\sigma_1, \ldots, \circ r_{n-1}\sigma_{n-1}$
if $n < \alpha$ **then return** $D_{VK}\langle h, 1\rangle$;
else
  $\hat{r}_n \leftarrow$ Random ;
  $\boxed{\hat{r} \leftarrow \text{ Random}}$;
  $\boxed{\hat{u} \leftarrow y \cdot g^{\hat{r}} \cdot h(1, r_{n-\alpha+1}, \ldots, r_{n-1}, \hat{r}_n)}$;
  $\hat{\sigma}_n \leftarrow \sigma(\hat{u})$;
end
return $\hat{r}_n\hat{\sigma}_n$;
$D*_{VK}^{\alpha-1}\langle h, 1\rangle$ `is almost identical to` $D*_{VK}^{\alpha-1}\langle h, 1\rangle$`. We highlighted the differences.`

We set up a reduction algorithm that uses the stegosystem as a black box and controls the actions of $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$. The reduction would get a challenge string $y = g^x$, where g is a generator of the group G and x is unknown. Next, the reduction would ask the stegosystem to generate a stegotext. Whenever the stegosystem queries $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$, the reduction would redirect the call to $D**_{VK}^{\alpha-1}\langle\cdot,1\rangle$. Algorithm 5 shows how $D**_{VK}^{\alpha-1}\langle\cdot,1\rangle$ inserts y into every signature. $D**_{VK}^{\alpha-1}\langle\cdot,1\rangle$ ensures that the returned signature $\hat{\sigma}_n$ is valid only if $r_{n-\alpha} = \log_g(y \cdot g^{\hat{r}}) = \log_g(g^{x+\hat{r}}) = x + \hat{r}$, where $\hat{r}$ is chosen by $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$. Since the signature σ_i is generated by $D*_{VK}^{\alpha-1}\langle\cdot,1\rangle$, we know that $s_{i-\alpha} = x + \hat{r}$. The reduction outputs $s_{i-\alpha} - \hat{r}$, thereby calculating the discrete logarithm. As a result, the probability that this case occurs is $\mathbf{DL}(t + O(q), q, k)$.

Based on our case analysis, we see that $\mathbf{InSec}_{\mathcal{S},D_{VK}}^{\mathsf{cha}}(t, 1, n, k) \geq 1 - \mathbf{InSec}_\Sigma^{\mathsf{sig}}(t+O(1), q, k) - \mathbf{DL}(t+O(q), k)$. Substituting $n = \alpha + 1$ proves the first part of the lemma. Furthermore, we've shown that $\forall i \geq 1$, the probability that an arbitrary signature σ_i is valid is at most $\mathbf{InSec}_\Sigma^{\mathsf{sig}}(t + O(1), q, k) + \mathbf{DL}(t + O(q), k)$. □

Theorem 5.6. Assume a stegosystem $\mathcal{S}$ has a database of q_1 covertexts generated by $D_{VK}\langle\lambda, n\rangle$ and the ability to query $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$ q_2 times. We can create an adversary A that distinguishes the output of D_{VK} from $\mathcal{S}$. A gets VK as input and permission to query a mystery oracle that is either D_{VK} or $\mathcal{S}$. A will ask its oracle to generate $q_1 + 1$ covertexts of length N. A outputs 1 if the oracle returns any duplicate covertexts or any invalid covertexts. If the oracle is $D_{VK}\langle\lambda,\cdot\rangle$, then A outputs 1 with probability $\Gamma(N, N, q_1 + 1, k)$ (the probability that duplicate covertexts occur). We examine what happens when the oracle is $\mathcal{S}$.

$\mathcal{S}$ can use its covertext database to generate stegotexts. Each covertext of length n can generate at most 1 valid stegotext of length N (the stegosystem can take an N document prefix). The stegosystem cannot take an arbitrary substring of a covertext because it would have to forge a signature on the new first integer and the α dummy arguments.

$\mathcal{S}$ gives A a list of q_1+1 stegotexts: $s^{(1)}, \ldots, s^{(q_1+1)}$. Each stegotext $s^{(i)}$ can be parsed as $r_1^{(i)}\sigma_1^{(i)} \circ \cdots \circ r_N^{(i)}\sigma_N^{(i)}$. $\mathcal{S}$ can easily create q_1 distinct stegotexts from its covertext dictionary. We examine how $\mathcal{S}$ generates the q_1+1st stegotext. There are 3 cases:

1. $\mathcal{S}$ has generated a new message signature pair that is not in the covertext database and that did not come from $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$. Then $\mathcal{S}$ has broken the security of the signature scheme Σ. $\mathcal{S}$ ran in $(q_1+1)t$ time and made nq_1+q_2 queries to $\sigma(SK,\cdot)$ (via its queries to $D_{VK}\langle\lambda,\cdot\rangle$ and $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$). Therefore, this case occurs with probability at most $\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}((q_1+1)t, nq_1+q_2, k)$.

2. $\mathcal{S}$ used a signature generated by $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$. By Lemma 5.9, we know that $\forall i,j > \alpha$, $\mathcal{S}$ can use $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$ to generate a valid $\sigma_j^{(i)}$ with probability at most $\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t+O(1), q_2, k)+\mathbf{DL}(t+O(q_2),k)$. Therefore, the probability that this case occurs is the total number of such signatures $(N-\alpha)(q_1+1)$ times the probability that any particular one was generated by $D_{VK}^{\alpha-1}\langle\cdot,1\rangle$. This gives a total probability of: $(N-\alpha)(q_1+1)(\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t+O(1),q_2,k)+\mathbf{DL}(t+O(q_2),k))$

3. The covertext database contains two sequences of α integers, thus letting $\mathcal{S}$ cut and paste two quotes. This occurs with probability $\Gamma(\alpha,n,q_2,k)$ (see Definition 5.4).

Adding up the probabilities from the case analysis above, we get that

$$\begin{aligned}\mathbf{Adv}_{\mathcal{S},D_{VK}}^{\mathsf{cha}}(A,k) \geq 1 &- \Gamma(N,N,q_1+1,k) - \mathbf{InSec}_{\Sigma}^{\mathsf{sig}}((q_1+1)t, nq_1+q_2, k)\\ &- (N-\alpha)(q_1+1)(\mathbf{InSec}_{\Sigma}^{\mathsf{sig}}(t+O(1),q_2,k)+\mathbf{DL}(t+O(q_2),k))\\ &- \Gamma(\alpha,n,q_2,k)\end{aligned}$$

A runs in $O((t+N)(q_1+1))$ time and makes q_1+1 queries of total length $N(q_1+1)$. Therefore, $\mathbf{InSec}_{\mathcal{S},D_{VK}}^{\mathsf{cha}}(O((t+N)(q_1+1)), q_1+1, N(q_1+1)) \geq \mathbf{Adv}_{\mathcal{S},D_{VK}}^{\mathsf{cha}}(A,k) \geq 1-\nu(k)$ for the negligible function ν defined above. This gives us the lower bound of $1-\nu(k)$ on the insecurity of $\mathcal{S}$. □

6 Conclusion

Our results link current theoretical research to real world stegosystems. We show that a stegosystem must assume that its approximation of the covertext distribution is correct. A slight error, or a missed correlation, can lead to almost certain detection. It is impossible to leverage incomplete or incorrect information to somehow create properly distributed covertexts. However, our work shows how to test the accuracy of the information a stegosystem does have. From our definition of α-memoryless, we see that all a stegosystem needs is a sampler that can generate a *single* document correctly based on a *randomly chosen* covertext.

References

[AP98] Ross J. Anderson and Fabien AP Petitcolas. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481, May 1998.

[BC05] Michael Backes and Christian Cachin. Public-key steganography with active attacks. In Joe Kilian, editor, *Theory of Cryptography Conference Proceedings*, volume 3378 of *LNCS*, pages 210–226. Springer Verlag, 2005.

[Cac98] Christian Cachin. An information-theoretic model for steganography. In David Aucsmith, editor, *Proc. 2nd Information Hiding Workshop*, volume 1525 of *LNCS*, pages 306–318. Sprinter Verlag, 1998.

[DIRR05] Nenad Dedić, Gene Itkis, Leonid Reyzin, and Scott Russell. Upper and lower bounds on black-box steganography. In Joe Kilian, editor, *Theory of Cryptography Conference Proceedings*, volume 3378 of *LNCS*, pages 227–244. Springer Verlag, 2005.

[Gol04] Oded Goldreich. Foundations of cryptography: Volume 2, basic applications. 2004.

[HLvA02] Nicholas J. Hopper, John Langford, and Louis von Ahn. Provably secure steganography. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *LNCS*. Springer, 2002.

[Hop04] Nicholas J. Hopper. Toward a theory of steganography. CMU Ph.D. Thesis, 2004.

[Hop05] Nicholas J. Hopper. On steganographic chosen covertext security. In *ICALP 2005, 32nd Annual International Colloquium on Automata, Languages and Programming, Lisboa, Portugal, July 11-15 2005, Proceedings*, 2005.

[Le03] Tri Van Le. Efficient provably secure public key steganography. Technical report, Florida State University, 2003. Cryptography ePrint Archive, `http://eprint.iacr.org/2003/156`.

[LK03] Tri Van Le and Kaoru Kurosawa. Efficient public key steganography secure against adaptively chosen stegotext attacks. Technical report, Florida State University, 2003. Cryptography ePrint Archive, `http://eprint.iacr.org/2003/244`.

[LM04] Anna Lysyanskaya and Mira Meyerovich. Steganography with imperfect sampling. At: CRYPTO 2004 Rump Session, August 2004, 2004.

[MLC01] Ira S. Moskowitz, Garth E. Longdon, and LiWu Chang. A new paradigm hidden in steganography. In *Proceedings of the 2000 workshop on New Security Paradigms*. ACM Press, 2001.

[PKSM] Kyle Petrowski, Mehdi Kharrazi, Husrev T. Sencar, and Nasir Memon. Psteg: steganographic embedding through patching. In *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[RR03] Leonid Reyzin and Scott Russell. Simple stateless steganography. Technical Report ePrint Archive 2003/093, Boston University, 2003. Cryptography ePrint Archive, from `http://eprint.iacr.org/2003/093`.

[Sal03] Phil Sallee. Model-based steganography. In *IWDW*, pages 154–167, 2003.

[vAH04] Louis von Ahn and Nicholas J. Hopper. Public-key steganography. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 323–341. Springer Verlag, 2004.