Public-Key Steganography

Luis von Ahn and Nicholas J. Hopper

Carnegie Mellon University

Abstract. Informally, a public-key steganography protocol allows two parties, who have never met or exchanged a secret, to send hidden messages over a public channel so that an adversary cannot even detect that these hidden messages are being sent. Unlike previous settings in which provable security has been applied to steganography, public-key steganography is information-theoretically *impossible*. In this work we introduce computational security conditions for public-key steganography similar to those introduced by Hopper, Langford and von Ahn [13] for the private-key setting. We also give the first protocols for public-key steganography and steganographic key exchange that are provably secure under standard cryptographic assumptions. Additionally, in the random oracle model, we present a protocol that is secure against adversaries that have access to a decoding oracle (the steganographic equivalent of CCA-2 adversaries).

Key Words: Steganography, Public-Key Cryptography, Provable Security

1 Introduction

Steganography refers to the problem of sending messages hidden in "innocent-looking" communications over a public channel, so that an adversary eavesdropping on the channel cannot even detect the presence of the hidden messages. Simmons [23] gave the most popular formulation of the problem: two prisoners, Alice and Bob, wish to plan an escape from jail. However, the prison warden, Ward, can monitor any communication between Alice and Bob, and if he detects any hint of "unusual" communications, he throws them both in solitary confinement. Alice and Bob must then transmit their secret plans so that nothing in their communication seems "unusual" to Ward.

There have been many proposed solutions to this problem, ranging from rudimentary schemes using invisible ink [14] to a protocol which is provably secure assuming that one-way functions exist [13]. However, the majority of these protocols have focused on the case where Alice and Bob share a secret or private key. If Alice and Bob were incarcerated before the need for steganography arose, these protocols would not help them. In contrast, public-key steganography allows parties to communicate steganographically with no prior exchange of secrets. As with public-key encryption, the sender of a message still needs to know the recipient's public key or otherwise participate in a key exchange protocol. While it is true that if there is no global PKI, the use of public keys might raise suspicion, in many cases it is the sender of a message who is interested in concealing his communication and there is no need for him to publish any keys.

In this paper we consider the notion of public-key steganography against adversaries that do not attempt to disrupt the communication between Alice and Bob (i.e., the goal of the adversary is only to detect whether steganography is being used and not to disrupt the communication between the participants). We show that secure public-key steganography exists if any of several standard cryptographic assumptions hold (each of these assumptions implies semantically secure public-key cryptography). We also show that secure steganographic key exchange is possible under the Integer Decisional Diffie-Hellman (DDH) assumption. Furthermore, we introduce a protocol that is secure in the random oracle model against adversaries that have access to a decoding oracle (the steganographic equivalent of CCA-2 adversaries).

Related Work. There has been very little work work on provably secure steganography (either in the private or the public key settings). A critical first step in this field was the introduction of an information-theoretic model for steganography by Cachin [5], and several papers have since given similar models [16, 19, 26]. Unfortunately, these works are limited in the same way that information-theoretic cryptography is limited. In particular, in any of these frameworks, secure steganography between two parties with no shared secret is impossible. Hopper, Langford, and von Ahn [13] have given a theoretical framework for steganography based on computational security. Our model will be substantially similar to theirs, but their work addresses only the shared-key setting, which is already possible information-theoretically. Although one of their protocols can be extended to the public-key setting, they do not consider formal security requirements for public-key steganography, nor do they consider the notions of steganographic-key exchange or adversaries that have access to both encoding and decoding oracles.

Anderson and Petitcolas [1], and Craver [9], have both previously described ideas for public-key steganography. This work will differ from theirs in several significant ways:

1. [1] and [9] do not attempt to give rigorous definitions, and give only heuristic arguments for the security of their constructions. In contrast, we will give rigorous definitions and proofs of security.
2. [1] does not describe any mechanism for generating encoded messages, but simply assumes "the ability to manipulate some bits of the cover." Similarly, [9] assumes the existence of a "supraliminal function" F and the ability to generate an x for which $F(x) = y$, for arbitrary y. In contrast, our model is constructive and does not assume the existence of a function with non-standard properties.

Inspired by a previous version of our work, a recent IACR pre-print [24] attempts to give a provably secure public-key stegosystem. Unfortunately this work contains a flaw. The author of [24] claims that his stegosystem has probability zero of decoding error; while this is true in the restricted case that the channel distribution is known exactly by both the sender and recipient, it is easy to construct an (uncountably) infinite set of channels for which the general construction has correct decoding probability approaching zero. We do not know of a way to repair the construction, which in fact fails for many natural channels. Furthermore, [24] only considers a notion similar to our weakest security condition.

To the best of our knowledge, we are the first to provide a formal framework for public-key steganography and to *prove* that public-key steganography is possible (given that standard cryptographic assumptions hold). We are also the first to consider adversaries that have access to decoding oracles (in a manner analogous to CCA-2 adversaries); we show that security against such adversaries can be achieved in the random oracle model. We stress that our protocols are not robust against adversaries wishing to render the steganographic communication channel useless. Throughout the paper, the goal of the adversary is detection, not disruption.

2 Definitions

Preliminaries. A function $\mu : \mathbb{N} \to [0,1]$ is said to be *negligible* if for every $c > 0$, for all sufficiently large n, $\mu(n) < 1/n^c$. We denote the length (in bits) of a string or integer s by $|s|$. The concatenation of string s_1 and string s_2 will be denoted by $s_1||s_2$. We also assume the existence of efficient, unambiguous *pairing* and *un-pairing* operations, so (s_1, s_2) is not the same as $s_1||s_2$. We let U_k denote the uniform distribution on k bit strings. If X is a finite set, we let $U(X)$ denote the uniform distribution on X. If $\mathcal{C}$ is a distribution with finite support X, we define the *minimum entropy* of $\mathcal{C}$, $H_\infty(\mathcal{C})$, as

$$H_\infty(\mathcal{C}) = \min_{x \in X} \left\{ \log_2 \frac{1}{\Pr_{\mathcal{C}}[x]} \right\} .$$

We say that a function $f : X \to \{0,1\}$ is *ϵ-biased* if $|\Pr_{x \leftarrow \mathcal{C}}[f(x) = 0] - 1/2| < \epsilon$. We say f is *unbiased* if f is ϵ-biased for ϵ a negligible function of the appropriate security parameter. We say f is *perfectly unbiased* if $\Pr_{x \leftarrow \mathcal{C}}[f(x) = 0] = 1/2$.

Integer Decisional Diffie-Hellman. Let P and Q be primes such that Q divides $P-1$, let $\mathbb{Z}_P^*$ be the multiplicative group of integers modulo P, and let $g \in \mathbb{Z}_P^*$ have order Q. Let $\mathbf{A}$ be an adversary that takes as input three elements of $\mathbb{Z}_P^*$ and outputs a single bit. Define the *DDH advantage of* $\mathbf{A}$ *over* (g, P, Q) as:

$$\mathbf{Adv}^{\mathsf{ddh}}_{g,P,Q}(\mathbf{A}) = \left| \Pr_{a,b,r}[\mathbf{A}_r(g^a, g^b, g^{ab}) = 1] - \Pr_{a,b,c,r}[\mathbf{A}_r(g^a, g^b, g^c) = 1] \right| ,$$

where $\mathbf{A}_r$ denotes the adversary $\mathbf{A}$ running with random tape r, a, b, c are chosen uniformly at random from $\mathbb{Z}_Q$ and all the multiplications are over $\mathbb{Z}_P^*$. Define *the DDH insecurity of* (g, P, Q) as $\mathbf{InSec}^{\mathsf{ddh}}_{g,P,Q}(t) = \max_{\mathbf{A} \in \mathcal{A}(t)} \left\{ \mathbf{Adv}^{\mathsf{ddh}}_{g,P,Q}(\mathbf{A}) \right\}$, where $\mathcal{A}(t)$ denotes the set of adversaries $\mathbf{A}$ that run for at most t time steps.

Trapdoor One-way Permutations. A trapdoor one-way permutation family Π is a sequence of sets $\{\Pi_k\}_k$, where each Π_k is a set of bijective functions $\pi : \{0,1\}^k \to \{0,1\}^k$, along with a triple of algorithms (G, E, I). $G(1^k)$ samples an element $\pi \in \Pi_k$ along with a *trapdoor* τ; $E(\pi, x)$ evaluates $\pi(x)$ for $x \in \{0,1\}^k$; and $I(\tau, y)$ evaluates $\pi^{-1}(y)$. For a PPT $\mathbf{A}$ running in time $t(k)$, denote the advantage of $\mathbf{A}$ against Π by

$$\mathbf{Adv}^{\mathsf{ow}}_{\Pi}(\mathbf{A}, k) = \Pr_{(\pi,\tau) \leftarrow G(1^k), x \leftarrow U_k}[\mathbf{A}(\pi(x)) = x] .$$

Define the insecurity of Π by $\mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t, k) = \max_{\mathbf{A} \in \mathcal{A}(t)} \{\mathbf{Adv}^{\mathsf{ow}}_{\Pi}(\mathbf{A}, k)\}$, where $\mathcal{A}(t)$ denotes the set of all adversaries running in time $t(k)$. We say that Π is a trapdoor one-way permutation family if for every probabilistic polynomial-time (PPT) $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{ow}}_{\Pi}(\mathbf{A}, k)$ is negligible in k.

Public-Key Encryption Indistinguishable From Random Bits. We will require public-key encryption schemes that are secure in a slightly non-standard model, which we will denote by IND\$-CPA in contrast to the more standard IND-CPA. Let $\mathcal{E} = (G, E, D)$ be a probabilistic public-key encryption scheme, where $E : \mathcal{PK} \times \mathcal{R} \times \mathcal{P} \to \mathcal{C}$. Consider a game in which an adversary $\mathbf{A}$ is given access to an oracle which is either:

- E_{PK} for $(PK, SK) \leftarrow G(1^k)$; that is, an oracle which given a message m, uniformly selects random bits r and returns $E_{PK}(r, m)$; or
- $\$(\cdot) = U_{|E_{PK}(\cdot)|}$; that is, an oracle which on any query ignores its input and returns a uniformly selected output of the appropriate length.

$\mathbf{A}$ is also given access to the public key PK used by its oracle to answer queries. Let $\mathcal{A}(t, q, l)$ be the set of adversaries $\mathbf{A}$ which make $q(k)$ queries to the oracle totalling at most $l(k)$ bits and run for $t(k)$ time steps. Define the IND\$-CPA advantage of $\mathbf{A}$ against $\mathcal{E}$ as

$$\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) = \left| \Pr_{(PK,SK) \leftarrow G(1^k), r \leftarrow \{0,1\}^*}[\mathbf{A}_r^{E_{PK}}(PK) = 1] - \Pr_{(PK,SK),r}[\mathbf{A}_r^{\$}(PK) = 1] \right|$$

where $\mathbf{A}_r$ denotes the adversary $\mathbf{A}$ with random tape r, and the probabilities are also taken over the randomness of the oracles $E_{PK}, \$$. Define the insecurity of $\mathcal{E}$ as $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, q, l, k) = \max_{\mathbf{A} \in \mathcal{A}(t,q,l)} \left\{ \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) \right\}$. $\mathcal{E}$ is (t, q, l, k, ϵ)*-indistinguishable from random bits under chosen plaintext attack* if $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, q, l, k) \le \epsilon(k)$. $\mathcal{E}$ is called *indistinguishable from random bits under chosen plaintext attack* (IND\$-CPA) if for every probabilistic polnyomial-time (PPT) $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k)$ is negligible in k.

We note that using a family of trapdoor permutations on domain $\{0,1\}^k$, the Efficient Probabilistic Encryption scheme of [11] (generalized from the scheme of [3]) is IND\$-CPA secure. Thus, under the assumption that such families exist, IND\$-CPA public-key encryption also exists. In Appendix D, we show how to construct schemes satisfying this condition under more general cryptographic assumptions, and give direct constructions under popular concrete assumptions.

Existentially Unforgeable Digital Signature Schemes. Let $\mathcal{SG} = (G, S, V)$ be a digital signature scheme. Consider the following game that an adversary $\mathbf{A}$ plays against $\mathcal{SG}$: the adversary $\mathbf{A}$ is given VK and oracle access to S_{SK}, where $(SK, VK) \leftarrow G(1^k)$. $\mathbf{A}$ makes $q(k)$ oracle queries of at most $l(k)$ bits to get back $\{S_{SK}(M_1), ..., S_{SK}(M_q)\}$. $\mathbf{A}$ then outputs a pair (M, σ_M). $\mathbf{A}$ wins if $M \notin \{M_1, ..., M_q\}$ and $V(VK, M, \sigma_M) = 1$.

Denote the event of $\mathbf{A}_r$ winning the game by $win_s(\mathbf{A}_r, k)$, where r denotes the random coins used by $\mathbf{A}$, k is the security parameter used to generate the keys, and s denotes the randomness used by the game (in generating (SK, VK) and in generating the $q(k)$ signatures). Let $\mathcal{A}(t, q, l)$ be the set of adversaries $\mathbf{A}$ which make $q(k)$ queries to the oracle of at most $l(k)$ bits and run for $t(k)$ time steps. Define the EUF-CMA advantage of $\mathbf{A}$ against $\mathcal{SG}$ as

$$\mathbf{Adv}^{\mathsf{cma}}_{\mathcal{SG}}(\mathbf{A}, k) = \left| \Pr_{s,r \leftarrow \{0,1\}^*}[win_s(\mathbf{A}_r, k)] \right| .$$

Define the insecurity of $\mathcal{SG}$ as $\mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t, q, l, k) = \max_{\mathbf{A} \in \mathcal{A}(t,q,l)} \left\{ \mathbf{Adv}^{\mathsf{cma}}_{\mathcal{SG}}(\mathbf{A}, k) \right\}$. We say that $\mathcal{SG}$ is (t, q, l, k, ϵ)*-existentially unforgeable under chosen message attack* if $\mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t, q, l, k) \leq \epsilon(k)$. $\mathcal{SG}$ is called *existentially unforgeable under chosen message attack* (EUF-CMA) if for every PPT $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{cma}}_{\mathcal{SG}}(\mathbf{A}, k)$ is negligible in k. We note that EUF-CMA signature schemes exist if and only if one-way functions exist [17, 20].

3 Channels

We seek to define steganography in terms of indistinguishability from a "usual" or innocent-looking distribution on communications. In order to do so, we must characterize this innocent-looking distribution. We follow [13] in using the notion of a channel, which models a prior distribution on the entire sequence of communication from one party to another:

Definition. Let D be an efficiently recognizable, prefix-free set of strings, or *documents*. A *channel* is a distribution on sequences $s \in D^*$.[1]

Any particular sequence in the support of a channel describes one possible outcome of all communications from Alice to Bob. The process of drawing from the channel, which results in a *sequence* of documents, is equivalent to a process that repeatedly draws a single "next" document from a distribution consistent with the history of already drawn documents. Therefore, we can think of communication as a series of these partial draws from the channel distribution, conditioned on what has been drawn so far. Notice that this notion of a channel is more general than the typical setting in which every symbol is drawn independently according to some fixed distribution: our channel explicitly models the dependence between symbols common in typical real-world communications.

Let $\mathcal{C}$ be a channel. We let $\mathcal{C}_h$ denote the marginal channel distribution on a single document from D conditioned on the history h of already drawn documents; we let $\mathcal{C}_h^l$ denote the marginal distribution on sequences of l documents conditioned on h. When we write "sample $x \leftarrow \mathcal{C}_h$" we mean that a single document should be returned according to the distribution conditioned on h. We use $\mathcal{C}_{A \to B, h}$ to denote the channel distribution on the communication from party A to party B.

We will require that a channel satisfy a minimum entropy constraint for all histories. Specifically, we require that there exist constants $L > 0$, $b > 0$, $\alpha > 0$ such that for all $h \in D^L$, either $\Pr_{\mathcal{C}}[h] = 0$ or $H_\infty(\mathcal{C}_h^b) \geq \alpha$. If a channel does not satisfy this property, then it is possible for Alice to drive the information content of her communications to 0, so this is a reasonable requirement. We say that a channel satisfying this condition is L*-informative*, and if a channel is L-informative for all $L > 0$, we say it is *always informative*. Note that this definition implies an additive-like property of minimum entropy for marginal distributions, specifically, $H_\infty(\mathcal{C}_h^{lb}) \geq l\alpha$. For ease of

[1] Hopper, Langford and von Ahn [13] define a channel so that each document in a sequence drawn from the channel has a time associated with it. These (presumably sending) times must then be respected by their protocols. We omit this timing information for simplicity, but remark that in a situation where precise timing information is available, it can be incorporated into our protocols orthogonally, with no effect on our results.

exposition, we will assume channels are always informative in the remainder of this paper; however, our theorems easily extend to situations in which a channel is L-informative.

In our setting, each ordered pair of parties (P, Q) will have their own channel distribution $\mathcal{C}_{P\to Q}$. In these cases, we assume that among the legitimate parties, only party A has oracle access to marginal channel distributions $\mathcal{C}_{A\to B,h}$ for every other party B and history h. On the other hand, we will allow the adversary oracle access to marginal channel distributions $\mathcal{C}_{P\to Q,h}$ for every pair P, Q and every history h. This allows the adversary to learn as much as possible about any channel distribution but does not require any legitimate participant to know the distribution on communications from any other participant. We will assume that each party knows the history of communications it has sent and received from every other participant.

We will also assume that cryptographic primitives remain secure with respect to oracles which draw from the marginal channel distributions $\mathcal{C}_{A\to B,h}$. Thus channels which can be used to solve the hard problems that standard primitives are based on must be ruled out. In practice this is of little concern, since the existence of such channels would have previously led to the conclusion that the primitive in question was insecure.

4 Public-key Steganography

Definition 1. (Stegosystem) A public-key stegosystem is a triple of probabilistic algorithms $S = (SG, SE, SD)$. $SG(1^k)$ generates a key pair $(PK, SK) \in \mathcal{PK} \times \mathcal{SK}$. SE takes a (public) key $PK \in \mathcal{PK}$, a string $m \in \{0,1\}^*$ (the *hiddentext*), and a message history h. SE also has access to a channel oracle for some channel $\mathcal{C}$, which can sample from $\mathcal{C}_h$ for any h. $SE(PK, m, h)$ returns a sequence of documents $s_1, s_2, \ldots, s_l$ (the *stegotext*) from the support of $\mathcal{C}_h^l$. SD takes a (secret) key $SK \in \mathcal{SK}$, a sequence of documents $s_1, s_2, \ldots, s_l$, and a message history h, and returns a hiddentext m. Additionally, for every polynomial p there must exist a negligible μ such that

$$\forall m \in \{0,1\}^{p(k)} : \quad \Pr_{(PK,SK)\leftarrow SG(1^k)}[SD(SK, SE(PK, m, h), h) = m] \geq 1 - \mu(k)$$

where the randomization is also over any coin tosses of SE, SD, SG and the oracle to $\mathcal{C}_h$.

Remarks

1. SE will be allowed access to an oracle that can sample from the channel distribution $\mathcal{C}_h$. We stress that *SE need not know the exact probabilities of documents in $\mathcal{C}_h$*. This is important to mention, as it is unreasonable to assume that the probabilities in $\mathcal{C}_h$ are known, whereas anybody communicating can be thought of as an oracle for the channel distribution $\mathcal{C}_h$.
2. We emphasize the terminology: the secret message that Alice wants to send to Bob is called the *hiddentext*; documents from the channel are called *covertexts*, and documents that are output by SE are called *stegotexts*.
3. While in general Alice will need to remember the history of documents transmitted to Bob, it is most desirable if Bob is not required to store the history of documents he has received from Alice. Some of our protocols require Bob to store this history, but it is straightforward to rewrite them so that Bob need only remember a collision-intractable digest of the history. In this paper we are not concerned with attacks in which Ward attempts to disrupt the communication between Alice and Bob, so the dependence of decoding on accurate history is chiefly a storage concern.

Steganographic Secrecy

We will model a warden attacking a stegosystem as an efficient oracle machine which plays the following oracle-distinguishing game:

1. W is given access to an oracle which samples documents from the marginal channel distributions $\mathcal{C}_{A\to B,h}$ for any history h. (This oracle allows W to learn the *covertext* distribution on all communications.)

2. W is given access to a second oracle which is either ST_{atk} or CT_{atk}. The oracle ST (for StegoText) will model the case in which the pair Alice and Bob are communicating steganographically, while the oracle CT (for CoverText) will model the case in which they are not. The exact distributions over $ST_{\mathsf{atk}}, CT_{\mathsf{atk}}$ vary depending on the attack model, atk. Below we will specify these distributions for $\mathsf{atk} \in \{\mathsf{cha}, \mathsf{cxo}\}$. Both oracles respond to the null query with any public keys generated by SG.
3. In the end, W outputs a bit.

We define the *advantage of W against stegosystem S over channel $\mathcal{C}$ in attack model* atk by

$$\mathbf{Adv}^{\mathsf{atk}}_{S,\mathcal{C}}(W,k) = \left| \Pr_r[W_r^{\mathcal{C},ST_{\mathsf{atk}}}(1^k) = 1] - \Pr_r[W_r^{\mathcal{C},CT_{\mathsf{atk}}}(1^k) = 1] \right| ,$$

where the warden uses random bits r and the probabilities are also taken over the oracles. Define *the insecurity of S over channel $\mathcal{C}$* by $\mathbf{InSec}^{\mathsf{atk}}_{S,\mathcal{C}}(t,q,l,k) = \max_{W \in \mathcal{W}(t,q,l)} \left\{ \mathbf{Adv}^{\mathsf{atk}}_{S,\mathcal{C}}(W,k) \right\}$, where $\mathcal{W}(t,q,l)$ denotes the set of all adversaries which make at most $q(k)$ oracle queries totaling at most $l(k)$ bits, and running in time at most $t(k)$. For ease of notation, we will denote $W^{\mathcal{C}}$ by W for the remainder of the paper.

Definition 2. (Steganographic secrecy) A Stegosystem S is called (t,q,l,k,ϵ)*-steganographically secret against* atk (SS-ATK) for the channel $\mathcal{C}$ if $\mathbf{InSec}^{\mathsf{atk}}_{S,\mathcal{C}}(t,q,l,k) \leq \epsilon(k)$.

Chosen Hiddentext Attack (CHA). In a chosen hiddentext attack, W is allowed to supply hiddentexts to Alice, in hopes of detecting whether or not she is using steganography in her communications with Bob. Thus W hopes to distinguish, for as many chosen messages m as he needs, between the case that Alice's communications are drawn from $SE(PK_B, m, h)$ or from $\mathcal{C}_{A \to B,h}^{|SE(PK_B,m,h)|}$ (where all m and h are chosen by W). Formally, we define the following distributions on ST_{cha} and CT_{cha}:

- To sample from ST_{cha}: draw $(PK, SK) \leftarrow SG(1^k)$ once, and respond to all queries with $ST_{\mathsf{cha}}(m,h) \leftarrow SE(PK,m,h)$.
- To sample from CT_{cha}: draw $(PK, SK) \leftarrow SG(1^k)$ once, and respond to all queries with $CT_{\mathsf{cha}}(m,h) \leftarrow \mathcal{C}_h^{|SE(PK,m,h)|}$.

Thus, the key pair (PK, SK) is drawn once and fixed, and Ward is allowed to query his oracle with arbitrary (m, h) pairs. (Note that a chosen hiddentext attack is the steganographic equivalent of a chosen plaintext attack against an encryption scheme.)

Chosen Exactly-One Attack (CXO). In a chosen exactly-one attack, Ward may both submit hiddentexts to Alice for encoding, and submit stegotexts to Bob for decoding, but *he is prohibited from accessing both at the same time* (i.e., with the same history). This models a scenario in which Ward can inject messages onto a network, but cannot replace them; for example, this may be the situation if he is connected to the same shared access network as Bob, but does not control the medium. As in all of our attack models, Ward's goal is to determine whether the specific pair of Alice and Bob are communicating steganographically rather than whether Alice or Bob are using steganography at all. In the case that Ward asks Bob to decode stegotexts, his goal is to detect whether Bob is receiving steganography from Alice by watching Bob's reaction to stegotexts (allegedly from Alice) generated by Ward himself. If Bob is not receiving steganography from Alice, he will always output the distinguished message $\perp$, meaning "no reaction." If Bob is receiving steganography from Alice, he will decode the stegotexts.

We formally define the oracle distributions $ST_{\mathsf{cxo}}, CT_{\mathsf{cxo}}$ as follows: First, draw $(PK, SK) \leftarrow SG(1^k)$, and set $\phi = \{\}$. Respond to queries using these programs:

$ST_{\mathsf{cxo}}(b \in \{\mathtt{enc}, \mathtt{dec}\}, m, h)$
if (b $=$ $\mathtt{enc}$) then:
  Sample $s \leftarrow SE(PK, m, h)$
  Set $\phi = \phi \cup \{h\}$
  return s
else
  If $h \in \phi$ return “”
  else return $SD(SK, m, h)$
	$CT_{\mathsf{cxo}}(b \in \{\mathtt{enc}, \mathtt{dec}\}, m, h)$
	if (b $=$ $\mathtt{enc}$) then:
	  Sample $s \leftarrow \mathcal{C}_h^{\lvert SE(PK,m,h)\rvert}$
	  Set $\phi = \phi \cup \{h\}$
	  return s
	else
	  If $h \in \phi$ return “”
	  else return $\perp$

Note that $\mathbf{InSec}^{\mathsf{cha}}_{S,\mathcal{C}}(t, q, l, k) \leq \mathbf{InSec}^{\mathsf{cxo}}_{S,\mathcal{C}}(t, q, l, k)$, since any CHA warden can be emulated by a CXO warden making only $(\mathtt{enc}, m, h)$-queries.

SS-CXO is the strongest notion of security that we are able to achieve in the standard model. Since any stegotext encoded by Alice can be thought of as a “challenge stegotext,” (Ward's goal is only to detect that it is, in fact, a stegotext rather than a covertext) this condition is somewhat analogous to non-adaptive chosen ciphertext security (IND-CCA1) for public-key encryption. However, in the random oracle model and assuming the channel is efficiently sampleable (i.e., there exists a PPT that can sample from the channel for any history), we can achieve a security condition analogous to *adaptive* chosen ciphertext security (IND-CCA2). We outline this security condition as well as the construction in Section 7.

5 Steganographic Key Exchange

A natural alternative to public-key steganography is *steganographic key exchange*: Alice and Bob exchange a sequence of messages, indistinguishable from normal communication traffic, and at the end of this sequence they are able to compute a shared key. So long as this key is indistinguishable from a random key to the warden, Alice and Bob can proceed to use their shared key in a secret-key stegosystem. In this section, we will formalize this notion.

Definition 3. (Steganographic Key Exchange Protocol) A *steganographic key exchange protocol,* or SKEP, is a quadruple of efficient probabilistic algorithms $S_{KE} = (SE_A, SE_B, SD_A, SD_B)$. SE_A and SE_B take as input a security parameter 1^k and a string of random bits, and output a sequence of documents of length $l(k)$; SD_A and SD_B take as input a security parameter, a string of random bits, and a sequence of documents of length $l(k)$, and output an element of the key space $\mathcal{K}$. Additionally, these algorithms satisfy the property that there exists a negligible function $\mu(k)$ satisfying:

$$\Pr_{r_A, r_B}[SD_A(1^k, r_A, SE_B(1^k, r_B)) = SD_B(1^k, r_B, SE_A(1^k, r_A))] \geq 1 - \mu(k) \ .$$

We call the output of $SD_A(1^k, r_A, SE_B(1^k, r_B))$ the *result* of the protocol, we denote this result by $S_{KE}(r_A, r_B)$, and we denote by $\mathcal{T}_k(r_A, r_B)$ (for transcript) the pair $(SE_A(1^k, r_A), SE_B(1^k, r_B))$.

Alice and Bob perform a key exchange using S_{KE} by sampling private randomness r_A, r_B, asynchronously sending $SE_A(1^k, r_A)$ and $SE_B(1^k, r_B)$ to each other, and using the result of the protocol as a key. Notice that in this definition a SKEP must be an asynchronous single-round scheme, ruling out multi-round key exchange protocols. This is for ease of exposition only.

We remark that many *authenticated* cryptographic key exchange protocols require three flows without a public-key infrastructure. Our SKE scheme will be secure with only two flows because we won't consider the same class of attackers as these protocols; in particular we will not worry about active attackers who alter the communications between Alice and Bob, and so Diffie-Hellman style two-flow protocols are possible. This may be a more plausible assumption in the SKE setting, since an attacker will not even be able to detect that a key exchange is taking place, while cryptographic key exchanges are typically easy to recognize.

Let W be a warden running in time t. We define W's *SKE advantage against* S_{KE} on channels $\mathcal{C}_{A\to B}$ and $\mathcal{C}_{B\to A}$ with security parameter k by:

$$\mathbf{Adv}^{\mathsf{ske}}_{S_{KE},\mathcal{C}_{A\to B},\mathcal{C}_{B\to A}}(W,k) = \left| \Pr_{r_A,r_B}[W(\mathcal{T}_k(r_A,r_B),S_{KE}(r_A,r_B)) = 1] - \Pr_{\sigma_A,\sigma_B,K}[W(\sigma_A,\sigma_B,K) = 1] \right| ,$$

where $\sigma_A \leftarrow \mathcal{C}^{l(k)}_{A\to B,h_A}, \sigma_B \leftarrow \mathcal{C}^{l(k)}_{B\to A,h_B}$, and $K \leftarrow \mathcal{K}$. We remark that, as in our other definitions, W also has access to channel oracles $\mathcal{C}_{A\to B,h}$ and $\mathcal{C}_{B\to A,h}$. Let $\mathcal{W}(t)$ denote the set of all wardens running in time t. The *SKE insecurity of* S_{KE} *on channels* $\mathcal{C}_A$ *and* $\mathcal{C}_B$ *with security parameter* k is given by $\mathbf{InSec}^{\mathsf{ske}}_{S_{KE},\mathcal{C}_{A\to B},\mathcal{C}_{B\to A}}(t,k) = \max_{W\in\mathcal{W}(t)} \left\{ \mathbf{Adv}^{\mathsf{ske}}_{S_{KE},\mathcal{C}_{A\to B},\mathcal{C}_{B\to A}}(W,k) \right\}$.

Definition 4. (Secure Steganographic Key Exchange) A SKEP S_{KE} is said to be (t,ϵ)-*secure for channels* $\mathcal{C}_{A\to B}$ *and* $\mathcal{C}_{B\to A}$ if $\mathbf{InSec}^{\mathsf{ske}}_{S_{KE}}(t,k) \le \epsilon(k)$. S_{KE} is said to be secure if for all polynomials p, S_{KE} is $(p(k),\epsilon(k))$-secure for some negligible function ϵ.

6 Constructions

Most of our protocols build on the following construction, a generalization of Construction 2 in [13] and similar to a protocol given by Cachin [5]. Let $f : D \to \{0,1\}$ be a public function (recall that $\mathcal{C}$ is a distribution on sequences of elements of D). If f is is perfectly unbiased on $\mathcal{C}_h$ for all h, then the following encoding procedure, on uniformly distributed l-bit input c, produces output distributed exactly according to $\mathcal{C}^l_h$:

Construction 1. (Basic encoding/decoding routines)

Procedure `Basic_Encode`:
Input: bits $c_1,\ldots,c_l$, history h, bound k
for $i = 1\ldots l$ do
  Let $j = 0$
  repeat:
    sample $s_i \leftarrow \mathcal{C}_h$, increment j
  until $f(s_i) = c_i$ OR $(j > k)$
  set $h = h||s_i$
Output: $s_1, s_2, \ldots, s_l$

Procedure `Basic_Decode`:
Input: Stegotext $s_1, s_2, \ldots, s_l$
for $i = 1\ldots l$ do
  set $c_i = f(s_i)$
set $c = c_1||c_2||\cdots||c_l$.
Output: c

Note that for infinitely many $\mathcal{C}_h$ there is no perfectly unbiased function f. In appendix C, we prove Proposition 1, which together with Proposition 2, justifies our use of unbiased functions. The proof for Proposition 2 is straightforward and is omitted from the paper.

Proposition 1. *Any channel $\mathcal{C}$ which is always informative can be compiled into a channel $\mathcal{C}^{(k)}$ which admits an efficiently computable function f such that for any polynomial-length sequence $h_1,\ldots,h_n$ satisfying* $\Pr_{\mathcal{C}}[h_i] \neq 0$, $\left|\Pr[f(C^{(k)}_{h_i}) = 1] - \frac{1}{2}\right|$ *is negligible in k for all $1 \le i \le n$.*

Proposition 2. *If f is ϵ-biased on $\mathcal{C}_h$ for all h, then for any k and $s_1, s_2, \ldots, s_l$:*

$$|\Pr_{c\leftarrow U_l}[\texttt{Basic\_Encode}(c,h,k) = s_1,s_2,\ldots,s_l] - \Pr_{s\leftarrow\mathcal{C}^l_h}[s = s_1,s_2,\ldots,s_l]| \le \epsilon l \ .$$

Using the previous two propositions, we will show that public-key steganography is possible in any channel that is always informative. We note that procedure `Basic_Encode` has a small probability of failure: `Basic_Decode(Basic_Encode(`c,h,k`))` might not equal c. This probability of failure, however, is negligible in k.

6.1 Chosen Hiddentext security

Let $E_{PK}(\cdot)$ and $D_{SK}(\cdot)$ denote the encryption and decryption algorithms for a public-key cryptosystem $\mathcal{E}$ which is indistinguishable from random bits under chosen plaintext attack (IND\$-CPA). Let ℓ be the expansion function of $\mathcal{E}$, i.e., $|E_{PK}(m)| = \ell(|m|)$. The following procedures allow encoding and decoding of messages in a manner which is steganographically secret under chosen hiddentext attack for the channel distribution $\mathcal{C}$:

Construction 2. (Chosen Hiddentext Security)

Procedure `CHA_Encode`:	**Procedure** `CHA_Decode`:
Input: message m, history h, key PK	**Input:** stegotext $s_1, s_2, \ldots, s_l$, secret key SK
Let $c = E_{PK}(m)$	Let $c = \texttt{Basic\_Decode}(s_1, s_2, \ldots, s_l)$
Output: $\texttt{Basic\_Encode}(c, h, k)$	**Output:** $D_{SK}(c)$

Lemma 1. *Assume f is ϵ-biased on $\mathcal{C}_h$ for all h. For any warden $W \in \mathcal{W}(t, q, l)$, we can construct an* IND\$-CPA *adversary* $\mathbf{A}$ *where*

$$\mathbf{Adv}^{\mathsf{cha}}_{\texttt{CHA},\mathcal{C}}(W, k) \leq \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) + \ell(l)\epsilon.$$

The running time of $\mathbf{A}$ is at most $t + O(kl)$ (k is the security parameter of (PK, SK) and is used as a bound for `Basic_Encode`*), and $\mathbf{A}$ makes q encryption queries for a total of l bits of plaintext.*

Proof. We are given a program W for distinguishing between the output of Construction 2 and the ordinary traffic of the channel, and an oracle for sampling blocks from that channel. We construct a program $\mathbf{A}$ which plays the IND\$-CPA game: distinguishing an $E_{PK}(\cdot)$ oracle from a uniform oracle. $\mathbf{A}^O$ simply runs W, using the encoding procedure `CHA_Encode` with the oracle O in place of $E_{PK}(\cdot)$ to respond to encoding queries. Consider the two cases for O:

1. $O(m) = E_{PK}(m)$. Then the responses to W's queries will be identically distributed to stegotexts resulting from the normal use of Construction 2, so $\Pr[\mathbf{A}^{E_{PK}}(PK) = 1] = \Pr[W^{ST_{\mathsf{cha}}}(1^k) = 1]$.
2. $O(m) = U_{|E_{PK}(m)|}$ is chosen uniformly from strings of appropriate length. Then the documents returned by `CHA_Encode` will be identically distributed to those returned by `Basic_Encode`; thus $|\Pr[\mathbf{A}^{\$}(PK) = 1] - \Pr[W^{CT_{\mathsf{cha}}}(1^k) = 1]| \leq \ell(l)\epsilon$ by Proposition 2.

Combining the cases, we have $\mathbf{Adv}^{\mathsf{cha}}_{\texttt{CHA},\mathcal{C}}(W, k) \leq |\Pr[\mathbf{A}^{E_{PK}}(PK) = 1] - \Pr[\mathbf{A}^{\$}(PK) = 1]| + \ell(l)\epsilon = \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) + \ell(l)\epsilon$, as claimed.

Theorem 1. *If f is ϵ-biased on $\mathcal{C}_h$ for all h, then*

$$\mathbf{InSec}^{\mathsf{cha}}_{\texttt{CHA},\mathcal{C}}(t, q, l, k) \leq \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kl), q, l, k) + \ell(l)\epsilon\ .$$

6.2 Chosen Exactly-One security

Let $\mathcal{SG} = (G_S, S, V)$ be a EUF-CMA secure signature scheme, with signature key K_S and verification key K_V, and let $\mathcal{E} = (G, E, D)$ be a IND\$-CPA encryption scheme with public key PK and secret key SK. Let ℓ be the expansion function of $\mathcal{E}$ and let ℓ_σ be the length of signatures generated by $\mathcal{SG}$. Then the following construction yields a SS-CXO secure stegosystem from Alice to Bob, when Alice knows PK, K_S and Bob knows SK, K_V. Assume also that all keys are generated with security parameter k.

Construction 3. (Chosen Exactly-One Security)

Procedure `CXO_Encode`:	**Procedure** `CXO_Decode`:
Input: m, h, PK, K_S	**Input:** $s_1, s_2, \ldots, s_l$, h, SK, K_V
Let $c = E_{PK}(m, S_{K_S}(h, m))$	Let $c = \texttt{Basic\_Decode}(s_1, s_2, \ldots, s_l)$
Output: $\texttt{Basic\_Encode}(c, h, k)$	Let $(m, \sigma) = D_{SK}(c)$
	Output: if $V(K_V, (h, m), \sigma) = 1$ then m, else $\perp$

Theorem 2. *Assume f is ϵ-biased on $\mathcal{C}_h$ for all h. Then*

$$\mathbf{InSec}^{\mathsf{cxo}}_{\texttt{CXO},\mathcal{C}}(t, q, l, k) \leq \mathbf{InSec}^{\mathsf{cma}}_{\mathcal{SG}}(t + O(kl), q, l, k) + \mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t + O(kl), q, l + q\ell_\sigma, k) + \ell(l + q\ell_\sigma)\epsilon\ .$$

Proof. (Sketch) We consider the hybrid oracle H which answers encoding queries using `CXO_Encode` and answers all decoding queries with $\perp$. Distinguishing this hybrid from ST_{cxo} equates to distinguishing `CXO_Decode` from the constant oracle $\perp$ on some history h for which no query of the form $(\mathtt{enc}, *, h)$ has been made. This can only happen if a decoding query contains a signature on a (m, h) pair which was never signed by `CXO_Encode` (because no encoding queries were ever made with the history h). So, intuitively, distinguishing between H and ST_{cxo} requires forging a signature. Similarly, since both H and CT_{cxo} answer all `dec` queries by $\perp$, distinguishing between them amounts to a chosen-hiddentext attack, which by Lemma 1 would give an IND\$-CPA attacker for $\mathcal{E}$. The result follows by the triangle inequality. For the full details of the proof, see Appendix A.

The above construction is vulnerable to a more powerful but still quite reasonable adversary. In particular, if Ward is able to modify messages sent by Alice before they are received by Bob, he may detect steganographic communications between them. The attack works as follows: suppose Alice outputs the stegotext $s_1, \ldots, s_l$. Ward chooses document s'_1 such that $f(s'_1) = f(s_1)$ and submits $s'_1, s_2, \ldots, s_l$ to Bob. Now if Bob decodes with the same history as Alice encoded with, the result will be a valid hiddentext rather than $\perp$. (Notice that this won't give a CXO attack against `CXO_Decode` because in a CXO attack Ward must use a different history, which has not been signed by Alice.) In Section 7 we specify a third security security property, SS-CSA, which guarantees security against attacks of this type.

6.3 Steganographic Key Exchange

The idea behind behind the construction for steganographic key exchange is simple: let g generate $\mathbb{Z}_P^*$, let Q be a large prime with $P = rQ + 1$ and r coprime to Q, and let $\hat{g} = g^r$ generate the subgroup of order Q. Alice picks random values $a \in \mathbb{Z}_{P-1}$ uniformly at random until she finds one such that $g^a \mod P$ has its most significant bit (MSB) set to 0 (so that $g^a \mod P$ is uniformly distributed in the set of bit strings of length $|P| - 1$). She then uses `Basic_Encode` to send all the bits of $g^a \mod P$ except for the MSB (which is zero anyway). Bob does the same and sends all the bits of $g^b \mod P$ except the most significant one (which is zero anyway) using `Basic_Encode`. Bob and Alice then perform `Basic_Decode` and agree on the key value $\hat{g}^{ab}$:

Construction 4. (Steganographic Key Exchange)

Procedure $\mathtt{SKE\_Encode}_A$: **Input:** primes P, Q, h, $g \in \mathbb{Z}_P^*$ of order rQ repeat: sample $a \leftarrow U(\mathbb{Z}_{P-1})$ until MSB of $g^a \mod P$ equals 0 Let c_a = all bits of g^a except MSB **Output:** $\mathtt{Basic\_Encode}(c_a, h, k)$	**Procedure** $\mathtt{SKE\_Decode}_A$: **Input:** Stegotext $s_1, s_2, \ldots, s_l$, exponent a Let $c_b = \mathtt{Basic\_Decode}(s_1, s_2, \ldots, s_l)$ **Output:** $c_b^{ra} \mod P = \hat{g}^{ab}$

Lemma 2. *Let f be ϵ-biased on $\mathcal{C}_{A \to B, h_A}$ and $\mathcal{C}_{B \to A, h_B}$ for all h_A, h_B. Then for any warden $W \in \mathcal{W}(t)$, we can construct a DDH adversary $\mathbf{A}$ where $\mathbf{Adv}^{\mathsf{ddh}}_{\hat{g},P,Q}(\mathbf{A}) \geq \frac{1}{4}\mathbf{Adv}^{\mathsf{ske}}_{\mathsf{SKE}}(W, k) - \epsilon|P|$. The running time of $\mathbf{A}$ is at most $t + O(k|P|)$.*

Proof. (Sketch) Define $\hat{r}$ to be the least element such that $r\hat{r} = 1 \mod Q$. The algorithm $\mathbf{A}$ works as follows. Given elements $(\hat{g}^a, \hat{g}^b, \hat{g}^c)$ of the subgroup of order Q, we uniformly choose elements $k_a, k_b \leftarrow \mathbb{Z}_r$, and set $c_a = (\hat{g}^a)^{\hat{r}} g^{k_a Q}$, and $c_b = (\hat{g}^b)^{\hat{r}} g^{k_b Q}$. If $MSB(c_a) = MSB(c_b) = 0$, we then return $W(\mathtt{Basic\_Encode}(c_a, h_A, k), \mathtt{Basic\_Encode}(c_b, h_B, k), \hat{g}^c)$, otherwise we return 0. Notice that the key computed by `SKE_Decode` would be $c_a^{rb} = \left((\hat{g}^a)^{\hat{r}} g^{k_a Q}\right)^{rb} = (\hat{g}^{ab})^{r\hat{r}} g^{rQk_a b} = \hat{g}^{ab}$.

The decrease in W's advantage comes from the fact that $\mathbf{A}$ excludes some elements of $\mathbb{Z}_P^*$ by sampling to get the MSB = 0, but we never exclude more than 1/2 of the cases for either c_a or c_b. The $\epsilon|P|$ difference follows from Proposition 2 and the fact that c_a,c_b are uniformly distributed on $U_{|P|-1}$.

Theorem 3. *If f is ϵ-biased on $\mathcal{C}_{A\to B,h_A}$ and $\mathcal{C}_{B\to A,h_B}$ for all h_A, h_B, then*

$$\mathbf{InSec}^{\mathsf{ske}}_{\mathsf{SKE},\mathcal{C}}(t,k) \le 4\epsilon|P| + 4\mathbf{InSec}^{\mathsf{ddh}}_{\hat{g},P,Q}(t + O(k|P|))) \ .$$

7 Chosen-Stegotext Attacks

We now introduce a security condition analogous to adaptive chosen ciphertext security for encryption schemes (IND-CCA2), SS-CSA. We will construct a stegosystem which is SS-CSA-secure in the Random Oracle model, for any channel which is *efficiently sampleable*: that is, for which there exists a PPT $\mathtt{C}$ such that for all h, $\mathtt{C}(h; U_k)$ and $\mathcal{C}_h$ are indistinguishable.

Definition. We formally define the oracle distributions $ST_{\mathsf{csa}}, CT_{\mathsf{csa}}$ as follows: First, draw $(PK, SK) \leftarrow SG(1^k)$, and set $\phi = \{\}$. Respond to queries using these programs:

ST_{csa} $(b \in \{\mathtt{enc}, \mathtt{dec}\}, m, h)$	CT_{csa} $(b \in \{\mathtt{enc}, \mathtt{dec}\}, m, h)$
if(b = $\mathtt{enc}$) then:	if(b = $\mathtt{enc}$) then:
Sample $s \leftarrow SE(PK, m, h)$	Sample $s \leftarrow \mathcal{C}_h^{\lvert SE(PK,m,h)\rvert}$
Set $\phi = \phi \cup \{(s, h)\}$	Set $\phi = \phi \cup \{(s, h)\}$
return s	return s
else	else
If $(m, h) \in \phi$ return “”	If $(m, h) \in \phi$ return “”
else return $SD(SK, m, h)$	else return $\perp$

Thus, in a chosen-stegotext attack, Ward may ask Alice to encode any (message, history) pair of his choosing, as often as he likes, and may query Bob on any pair (s, h) where s was not a result of an encoding query for history h. Notice that if Ward can ask Bob to decode any message output by Alice for the same history it was encoded with, he can detect the use of steganography between Alice and Bob; this is why we do not allow Ward to query Bob on such stegotexts. This restriction is roughly analogous to the standard restriction that an adaptive chosen-ciphertext attacker may not query his decryption oracle on the challenge ciphertext. Advantage and insecurity for SS-CSA are defined analogously to SS-CXO, except that we count encoding and decoding queries separately (as q_e and q_d) as well as counting the number of queries made to random oracles.

Construction. We assume that π_A, π_B are elements of trapdoor one-way permutation family Π_k, where Alice knows π_A^{-1} and Bob knows π_B^{-1}. In addition, we assume all parties have access to random oracles $F : \{0,1\}^* \to \{0,1\}^k$, $G : \{0,1\}^* \to \{0,1\}^k$, $H_1 : \{0,1\}^k \to \{0,1\}^*$, and $H_2 : \{0,1\}^* \to \{0,1\}^k$. The following construction slightly modifies techniques from [4], using the random oracles H_1 and H_2 with π_B to construct a pseudorandom non-malleable encryption scheme and the oracle F in conjunction with π_A to construct a strongly unforgeable signature scheme.

Construction 5. (Chosen Stegotext Security)

Procedure $\mathtt{UEncode}^G$:	**Procedure** $\mathtt{CSA\_Encode}^{F,G,H}$:	**Procedure** $\mathtt{CSA\_Decode}^{F,G,H}$:
Input: $c \in \{0,1\}^l$, $r \in \{0,1\}^k$, h	**Input:** $m_1 \cdots m_\ell$, h, π_A^{-1}, π_B	**Input:** $s_1, \ldots, s_l$, h, π_A, π_B^{-1}
for $i = 1 \ldots l$ do	Choose $r \leftarrow U_k$	Let $c = \mathtt{Basic\_Decode}(s_1, \ldots, s_l)$
Let $j = 0$	Let $\sigma = \pi_A^{-1}(F(r, m, h))$	Parse c as $y\|e\|\tau$.
repeat:	Let $e = H_1(r) \oplus (m, \sigma)$	Set $r = \pi_B^{-1}(y)$.
set $s_i = \mathtt{C}(h; G(h, r, c, j))$	Let $\tau = H_2(r, m, h)$	If $s \neq \mathtt{UEncode}^G(c, r, h)$ return $\perp$.
increment j	Let $y = \pi_B(r)$	Let $(m, \sigma) = e \oplus H_1(r)$
until $f(s_i) = c_i$ OR $(j > k)$	Let $c = y\|e\|\tau$	If $\tau \neq H_2(r, m, h)$ return $\perp$.
set $h = (h, s_i)$	**Output:** $\mathtt{UEncode}^G(c, r, h)$	If $\pi_A(\sigma) \neq F(r, m, h)$ return $\perp$.
Output: $s_1, s_2, \ldots, s_l$		**Output:** m

Theorem 4. *If f is ϵ-biased for $\mathcal{C}$, then*

$$\mathbf{InSec}^{\mathsf{csa}}_{\mathtt{CSA},\mathcal{C}}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l, k) \leq (2q_e + q_F)\mathbf{InSec}^{\mathsf{ow}}_{\pi}(t', k) + (l + 3q_e k)\epsilon + (q_e^2 + 2q_d)/2^k \ ,$$

where $t' \leq t + (q_G + q_F + q_{H_1} + q_{H_2})(q_e + q_d)T_\pi + k(l + 3q_e k)T_{\mathtt{C}}$, T_π is the time to evaluate members of π, and $T_{\mathtt{C}}$ is the running time of $\mathtt{C}$.

Intuitively, this stegosystem is secure because the encryption scheme employed is non-malleable, the signature scheme is strongly unforgeable, and each triple of hiddentext, history, and random-bits has a unique valid stegotext, which contains a signature on (m, h, r). Thus any adversary making a valid decoding query which was not the result of an encoding query can be used to forge a signature for Alice — that is, invert the one-way permutation π_A. The full proof is omitted for space considerations; see Appendix B for details.

We conjecture that the *cryptographic* assumptions used here can be weakened — in particular, a random oracle is not necessary given a public-key encryption scheme which satisfies IND\$-CPA and is non-malleable[2], and a signature scheme which is *strongly unforgeable*. However, it seems challenging to prevent our motivating attack without assuming ability to efficiently sample the channel.

8 Discussion and Open Problems

Need for a PKI. A potential stumbling block for public-key steganography is the need for a system which allows Alice and Bob to publish public keys for encryption and signatures without raising suspicion. The most likely source of a resolution to this issue is the existence of a global public-key infrastructure which publishes such public keys for every party in any case. In many cases (those modeled by the chosen hiddentext attack), however, it may be Alice who is trying to avoid suspicion while it is Bob who publishes the public key. For example Alice may be a government employee who wishes to leak a story and Bob a newspaper reporter, who may publish his public key daily.

In case Alice and Bob are both trying to avoid suspicion, it may be necessary to perform SKE instead. Even in this case, there is a need for a one-bit "secret channel" which alerts Bob to the fact that Alice is attempting key exchange. However, as long as Bob and Alice assume key exchange is occurring, it is easy to check at completion that it has indeed ocurred by using `Basic_Encode` to exchange the messages $F_K(A, h_A), F_K(B, h_B)$ for F a pseudorandom function.

Stegosystems with backdoors. Suppose we wish to design steganography software which will be used as a black box by many users. Then as long as there is some entropy in the stegosystem of choice, we can use public-key steganography to implement a backdoor into the stegosystem which is provably undetectable via input/output behavior, by using the encoding routine as an oracle for Construction 2, with a fixed hiddentext (1^k, for instance). This will make it possible, with enough intercepted messages, to detect the use of the steganography software. If a total break is desired and the software implements private-key steganography, we can replace 1^k by the user's private key.

Relationship to PKC: Complexity-theoretic implications. The definition of a SS-CHA secure public-key stegosystem already implies semantic security, so we have that if secure public-key stegosystems exist, then secure public-key cryptography exists; and likewise it is clear that SS-CSA security implies non-malleability. In contrast to the private-key results of [13], we are

[2] We are unaware of an encryption scheme in the standard model satisfying this requirement: nonmalleable encryption schemes following the Naor-Yung paradigm [18, 10, 21, 15] are easily distinguishable from random bits, and the schemes of Cramer and Shoup [7, 8] all seem to generate ciphertexts which are elements of recognizable subgroups.

not aware of a general result showing that the existence of any semantically secure public-key cryptosystem implies the existence of secure public-key steganography. However, our results allow construction of provably secure public-key steganography based on the security of any popular public-key cryptosystem.

References

1. R. J. Anderson and F. A. P. Petitcolas. On The Limits of Steganography. *IEEE Journal of Selected Areas in Communications*, 16(4), pages 474-481. 1998.
2. R. J. Anderson and F. A. P. Petitcolas. Stretching the Limits of Steganography. *Proceedings of the first International Information Hiding Workshop*, Springer LNCS 1174, pages 39-48. 1996.
3. M. Blum and S. Goldwasser. An Efficient Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information. *Advances in Cryptology: CRYPTO '84*, Springer LNCS 196, pages 289-302. 1985.
4. M. Bellare and P. Rogaway. Random Oracles are Practical. *Computer and Communications Security: Proceedings of ACM CCS 93*, pages 62–73, 1993.
5. C. Cachin. An Information-Theoretic Model for Steganography. *Proceedings of the Second International Information Hiding Workshop*, Springer LNCS 1525, pages 306-318. 1998.
6. R. Canetti, U. Feige, O. Goldreich and M. Naor. Adaptively Secure Multi-party Computation. *28th Symposium on Theory of Computing (STOC '96)*, pages 639-648. 1996.
7. R. Cramer and V. Shoup. A practical public-key cryptosystem provably secure against adaptive chosen ciphertext attack. *Advances in Cryptology: CRYPTO '98*, Springer LNCS 1462, pages 13-25. 1998.
8. R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. *Advances in Cryptology: EUROCRYPT '02*, Springer LNCS 2332, pages 45-64. 2002.
9. S. Craver. On Public-key Steganography in the Presence of an Active Warden. *Proceedings of the Second International Information Hiding Workshop*, Springer LNCS 1525, pages 355-368. 1998.
10. D. Dolev, C. Dwork, and M. Naor. Non-malleable Cryptography. *23rd Symposium on Theory of Computing (STOC '91)*, pages 542-552. 1991.
11. S. Goldwasser and M. Bellare. Lecture Notes on Cryptography. Unpublished manuscript, August 2001. available electronically at `http://www-cse.ucsd.edu/~mihir/papers/gb.html`.
12. J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom generator from any one-way function. *SIAM Journal of Computing*, 28(4), pages 1364-1396. 1999.
13. N. J. Hopper, J. Langford, and L. von Ahn. Provably Secure Steganography. *Advances in Cryptology: CRYPTO '02*, Springer LNCS 2442, pages 77-92. 2002.
14. D. Kahn. *The Code Breakers*. Macmillan 1967.
15. Y. Lindell. A Simpler Construction of CCA2-Secure Public Key Encryption. *Advances in Cryptology: EUROCRYPT '03*, Springer LNCS 2656, pages 241-254. 2003.
16. T. Mittelholzer. An Information-Theoretic Approach to Steganography and Watermarking. *Proceedings of the Third International Information Hiding Workshop*, Springer LNCS 1768, 2000.
17. M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. *21st Symposium on Theory of Computing (STOC '89)*, pages 33-43. 1989.
18. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. *22nd Symposium on Theory of Computing (STOC '90)*, pages 427-437. 1990.
19. J. A. O'Sullivan, P. Moulin, and J. M. Ettinger Information-theoretic analysis of Steganography. *Proceedings ISIT '98*. 1998.
20. J. Rompel. One-way functions are necessary and sufficient for secure signatures. *22nd Symposium on Theory of Computing (STOC '90)*, pages 387-394. 1990.
21. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. *40th IEEE Symposium on Foundations of Computer Science (FOCS '99)*, pages 543-553. 1999.
22. V. Shoup. A proposal for an ISO standard for public key encryption. Available electronically: `http://shoup.net/papers/iso-2_1.pdf`.
23. G. J. Simmons. The Prisoner's Problem and the Subliminal Channel. *Advances in Cryptology: CRYPTO' 83*, pages 51-67. 1983.
24. T. Van Le. Efficient Proven Secure Public Key Steganography. Cryptology ePrint Archive, Report 2003/156, September 3, 2003. Available electronically: `http://eprint.iacr.org/2003/156`.
25. A. Young and M. Yung. Kleptography: Using Cryptography against Cryptography. *Advances in Cryptology: EUROCRYPT '87*, Springer LNCS 1233, pages 62-74. 1987.
26. J. Zollner, H. Federrath, H. Klimant, A. Pftizmann, R. Piotraschke, A. Westfield, G. Wicke, G. Wolf. Modeling the security of steganographic systems. *Proceedings of the Second International Information Hiding Workshop*, Springer LNCS 1525, pages 344-354. 1998.

A Proof of Chosen Exactly-One security

Proof. Let $W \in \mathcal{W}(t,q,l)$. We will show that W must either forge a signature or distinguish the output of E from random bits. We will abuse notation slightly and denote $W^{ST_{\mathsf{cxo}}}$ by $W^{SE,SD}$, and $W^{CT_{\mathsf{cxo}}}$ by $W^{\mathcal{C},\perp}$. Then we have that

$$\mathbf{Adv}^{\mathsf{cxo}}_{\mathtt{CXO},\mathcal{C}}(W,k) = \left|\Pr[W^{SE,SD}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right| \ .$$

Consider the "hybrid" distribution which results by answering encoding queries using `CXO_Encode` but answering all decoding queries with $\perp$. (We denote this oracle by $(SE,\perp)$)

We construct a EUF-CMA adversary $\mathbf{A}_f$ which works as follows: given K_V, and a signing oracle for K_S, choose $(PK,SK) \leftarrow G_E(1^k)$; use the signing oracle and E_{PK}, D_{SK} to emulate `CXO_Encode` and `CXO_Decode` to W. If W ever makes a query to `CXO_Decode` which does not return $\perp$ then $\mathbf{A}_f$ halts and returns the corresponding $((m,h),\sigma)$ pair, otherwise $\mathbf{A}_f$ runs until W halts and returns $(0,0)$. If we let F denote the event that $W^{SE,SD}$ submits a valid decoding query to `CXO_Decode`, then we have that $\mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f) = \Pr[F]$.

We also construct a IND\$-CPA adversary $\mathbf{A}_d$ which works as follows: given an encryption oracle, choose $(K_S,K_V) \leftarrow G_S(1^k)$, use K_S and the encryption oracle to emulate `CXO_Encode` to W, and respond to any decoding queries with $\perp$. $\mathbf{A}_d$ returns the output of W. Note that $\mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l+q\ell_\sigma)\epsilon \geq \left|\Pr[W^{SE,\perp}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right|$, which follows from Theorem 1.

Then we have the following inequalities:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{cxo}}_{\mathtt{CXO},\mathcal{C}}(W) &= \left|\Pr[W^{SE,SD}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right| \\
&\leq \left|\Pr[W^{SE,SD}=1] - \Pr[W^{SE,\perp}=1]\right| + \left|\Pr[W^{SE,\perp}=1] - \Pr[W^{\mathcal{C},\perp}=1]\right| \\
&\leq \left|\Pr[W^{SE,SD}=1] - \Pr[W^{SE,\perp}=1]\right| + \mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l+q\ell_\sigma)\epsilon \\
&\leq \mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f) + \mathbf{Adv}^{\mathsf{cpa}}_{E}(\mathbf{A}_d) + \ell(l+q\ell_\sigma)\epsilon
\end{aligned}$$

Where the last line follows because (let D denote the event $W^{SE,SD}=1$, and notice that $D|\overline{F} \equiv W^{SE,\perp}=1$):

$$\begin{aligned}
\left|\Pr[D] - \Pr[W^{SE,\perp}=1]\right| &= \left|\Pr[D|F]\Pr[F] + \Pr[D|\overline{F}]\Pr[\overline{F}] - (\Pr[W^{SE,\perp}=1])\right| \\
&= \left|\Pr[D|F]\Pr[F] + \Pr[W^{SE,\perp}=1](1-\Pr[F]) - (\Pr[W^{SE,\perp}=1])\right| \\
&= \left|\Pr[F](\Pr[D|F] - \Pr[D|\overline{F}])\right| \\
&\leq \Pr[F] \\
&= \mathbf{Adv}^{\mathsf{cma}}_{(G_S,S,V)}(\mathbf{A}_f)
\end{aligned}$$

The theorem follows by the definition of insecurity, since both $\mathbf{A}_d$ and $\mathbf{A}_f$ run in time at most $t + O(kl)$.

B Proof of Chosen-Stegotext Security

We define the following sequence of hybrid oracle distributions:

1. $\mathsf{P0}(b,m,h) = CT_{\mathsf{csa}}$, the covertext oracle.
2. $\mathsf{P1}(b,m,h)$ responds to `dec` queries as in $\mathsf{P0}$, and responds to `enc` queries using $\texttt{CSA\_Encode}^{F,G,H}$ but with calls to $\texttt{UEncode}^G$ replaced by calls to `Basic_Encode`.
3. $\mathsf{P2}(b,m,h)$ responds to `dec` queires as in $\mathsf{P1}$, and responds to `enc` queries using $\texttt{CSA\_Encode}^{F,G,H}$.
4. $\mathsf{P3}(b,m,h) = ST_{\mathsf{csa}}$, the stegotext oracle.

We are given a CSA attacker $W \in \mathcal{W}(t, q_e, q_d, q_F, q_H, q_{H_1}, q_{H_2}, l)$ and wish to bound his advantage. Notice that $\mathbf{Adv}^{\mathsf{csa}}_{\mathsf{CSA},\mathcal{C}}(W,k) \le |\Pr[W^{P0}(1^k) = 1] - \Pr[W^{P1}(1^k)]| + |\Pr[W^{P1}(1^k) = 1] - \Pr[W^{P2}(1^k) = 1]| + |\Pr[W^{P2}(1^k) = 1] - \Pr[W^{P3}(1^k) = 1]|$. Hence, we can bound the advantage of W by the sum of its advantages in distinguishing the successive hybrids. For hybrids P,Q we will denote this advantage by $\mathbf{Adv}^{\mathsf{P},\mathsf{Q}}(W,k) = |\Pr[W^{\mathsf{P}}(1^k) = 1] - \Pr[W^{\mathsf{Q}}(1^k) = 1]|$.

Lemma 3. $\mathbf{Adv}^{\mathsf{P0},\mathsf{P1}}(W,k) \le q_e \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + 2^{-k}(q_e^2/2 - q_e/2) + (l + 3q_e k)\epsilon$

Proof. Assume WLOG that $\Pr[W^{P1}(1^k) = 1] > \Pr[W^{P0}(1^k) = 1]$. Let E_r denote the event that, when W queries $\mathsf{P1}$, the random value r never repeats, and let E_q denote the event that W never makes random oracle queries of the form $H_1(r)$ or $H_2(r,*,*)$ for an r used by $\texttt{CSA\_Encode}^{F,G,H}$, and let $E \equiv E_r \wedge E_q$. Then:

$$\begin{aligned}
\Pr[W^{P1}(1^k) = 1] - \Pr[W^{P0}(1^k) = 1] &= \Pr[W^{P1}(1^k) = 1|E](1 - \Pr[\overline{E}]) + \Pr[W^{P1}(1^k) = 1|\overline{E}]\Pr[\overline{E}] \\
&\quad - \Pr[W^{P0}(1^k) = 1] \\
&= \Pr[\overline{E}]\Big(\Pr[W^{P1}(1^k) = 1|\overline{E}] - \Pr[W^{P1}(1^k) = 1|E]\Big) \\
&\quad + \Big(\Pr[W^{P1}(1^k) = 1|E] - \Pr[W^{P0}(1^k) = 1]\Big) \\
&\le \Pr[\overline{E}] \quad + (l + 3q_e k)\epsilon \\
&\le \Pr[\overline{E_r}] + \Pr[\overline{E_q}] + (l + 3q_e k)\epsilon \\
&\le 2^{-k}\frac{q_e(q_e - 1)}{2} + \Pr[\overline{E_q}] + (l + 3q_e k)\epsilon \ ,
\end{aligned}$$

because if r never repeats and W never queries $H_1(r)$ or $H_2(r,*,*)$ for some r used by $\texttt{CSA\_Encode}^{F,G,H}$, then W cannot distinguish between the ciphertexts passed to `Basic_Encode` and random bit strings.

It remains to bound $\Pr[\overline{E_q}]$. Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_B which is given a value $\pi_B(x)$ and uses W in an attempt to find x, so that $\mathbf{A}$ succeeds with probability at least $(1/q_e)\Pr[\overline{E_q}]$. $\mathbf{A}$ picks (π_A, π_A^{-1}) from Π_k and i uniformly from $\{1, \ldots, q_e\}$, and then runs W answering all its oracle queries as follows:

- `enc` queries are answered as follows: on query $j \neq i$, respond using $\texttt{CSA\_Encode}^{F,G,H}$ but with calls to $\texttt{UEncode}^G$ replaced by calls to `Basic_Encode`. On the i-th query respond with $s = \texttt{Basic\_Encode}(\pi_B(x)||e_1||\tau_1, h)$ where $e_1 = h_1 \oplus (m, \sigma_1)$ and h_1, σ_1, τ_1 are chosen uniformly at random from the set of all strings of the appropriate length ($|e_1| = |m| + k$ and $|\tau_1| = k$), and set $\phi = \phi \cup \{(s,h)\}$.
- `dec` queries are answered using CT_{csa}.
- Queries to G, F, H_1 and H_2 are answered in the standard manner: if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length. If a query contains a value r for which $\pi_B(r) = \pi_B(x)$, halt the simulation and output r.

It should be clear that $\Pr[\mathbf{A}(\pi_B(x)) = x] \geq \frac{1}{q_e}(\Pr[\overline{E_q}])$.

Lemma 4. $\mathbf{Adv}^{\mathsf{P1,P2}}(W,k) \leq q_e \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + 2^{-k}(q_e^2/2 - q_e/2)$

Proof. Assume WLOG that $\Pr[W^{P2}(1^k) = 1] > \Pr[W^{P1}(1^k) = 1]$. Denote by E_r the event that, when answering queries for W, the random value r of $\texttt{CSA\_Encode}^{F,G,H}$ never repeats, and by E_q the event that W never queries $G(*, r, \pi_B(r)||*, *)$ for some r used by $\texttt{CSA\_Encode}^{F,G,H}$, and let $E \equiv E_r \wedge E_q$. Then:

$$\begin{aligned}
\Pr[W^{P2}(1^k) = 1] - \Pr[W^{P1}(1^k) = 1] &= \left(\Pr[W^{P2}(1^k) = 1|E]\Pr[E] + \Pr[W^{P2}(1^k) = 1|\overline{E}]\Pr[\overline{E}]\right) \\
&\quad - \left(\Pr[W^{P1}(1^k) = 1|E]\Pr[E] + \Pr[W^{P1}(1^k) = 1|\overline{E}]\Pr[\overline{E}]\right) \\
&= \Pr[\overline{E}]\left(\Pr[W^{P2}(1^k) = 1|\overline{E}] - \Pr[W^{P1}(1^k) = 1|\overline{E}]\right) \\
&\leq \Pr[\overline{E}] \\
&\leq 2^{-k}\frac{q_e(q_e-1)}{2} + \Pr[\overline{E_q}]
\end{aligned}$$

Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_B which is given a value $\pi_B(x)$ and uses W in an attempt to find x. $\mathbf{A}$ picks (π_A, π_A^{-1}) from Π_k and i uniformly from $\{1, \ldots, q_E\}$, and then runs W answering all its oracle queries as follows:

- $\texttt{enc}$ queries are answered as follows: on query $j \neq i$, respond using $\texttt{CSA\_Encode}^{F,G,H}$. On the i-th query respond with $s = \texttt{UEncode}^G(\pi_B(x)||e_1||\tau_1, r_1, h)$ where $e_1 = h_1 \oplus (m, \sigma_1)$ and $h_1, \sigma_1, \tau_1, r_1$ are chosen uniformly at random from the set of all strings of the appropriate length ($|e_1| = |m|+k$ and $|\tau_1| = k$), and set $\phi = \phi \cup \{(s,h)\}$.
- $\texttt{dec}$ queries are answered using CT_{csa}.
- Queries to G, F, H_1 and H_2 are answered in the standard manner: if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length. If a query contains a value r for which $\pi_B(r) = \pi_B(x)$, halt the simulation and output r.

It should be clear that $\Pr[\mathbf{A}(\pi_B(x)) = x] \geq \frac{1}{q_e}(\Pr[\overline{E_q}])$.

Lemma 5. $\mathbf{Adv}^{\mathsf{P2,P3}}(W,k) \leq q_F \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + q_d/2^{k-1} + q_e/2^k$

Proof. Given $W \in \mathcal{W}(t, q_e, q_d, q_F, q_G, q_{H_1}, q_{H_2}, l)$ we construct a one-way permutation adversary $\mathbf{A}$ against π_A which is given a value $\pi_A(x)$ and uses W in an attempt to find x. $\mathbf{A}$ chooses (π_B, π_B^{-1}) from Π_k and i uniformly from $\{1, \ldots, q_F\}$, and then runs W answering all its oracle queries as follows:

- $\texttt{enc}$ queries are answered using $\texttt{CSA\_Encode}^{F,G,H}$ except that σ is chosen at random and $F(r,m,h)$ is set to be $\pi_A(\sigma)$. If $F(r,m,h)$ was already set, fail the simulation.
- $\texttt{dec}$ queries are answered using $\texttt{CSA\_Decode}^{F,G,H}$, with the additional constraint that we reject any stegotext for which there hasn't been an oracle query of the form $H_2(r,m,h)$ or $F(r,m,h)$.
- Queries to G, F, H_1 and H_2 are answered in the standard manner (if the query has been made before, answer with the same answer, and if the query has not been made before, answer with a uniformly chosen string of the appropriate length) except that the i-th query to F is answered using $\pi_A(x)$.

$\mathbf{A}$ then searches all the queries that W made to the decryption oracle for a value σ such that $\pi_A(\sigma) = \pi_A(x)$. This completes the description of $\mathbf{A}$.

Notice that the simulation has a small chance of failure: at most $q_e/2^k$. For the rest of the proof, we assume that the simulation doesn't fail. Let E be the event that W makes a decryption query that is rejected in the simulation, but would not have been rejected by the standard $\texttt{CSA\_Decode}^{F,G,H}$. It is easy to see that $\Pr[E] \leq q_d/2^{k-1}$. Since the only way to differentiate P3 from P2 is by making a decryption query that P3 accepts but P2 rejects, and, conditioned on $\overline{E}$, this can only happen by inverting π_A on a some $F(r,m,h)$, we have that:

$$\mathbf{Adv}^{\mathsf{P2},\mathsf{P3}}(W,k) \leq q_F \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + q_d/2^{k-1} + q_e/2^k$$

C Negligibly biased functions for any channel

Our constructions require the existence of a function which is unbiased for $\mathcal{C}_h$ for every h which the warden W chooses. It is easy to see that for infinitely many channels, no such f exists. This is not a difficulty for our protocols, however, because we can compile any channel $\mathcal{C}$ into a new channel $\mathcal{C}^{(k)}$ which admits an efficient function which has bias negligible in k.

Let $l(k) = \omega(\log k)$. Then the channel $\mathcal{C}^{(k)}$ is simply a distribution on sequences of documents which are elements of $D^{l(k)}$ and the marginal distributions $\mathcal{C}_h^{(k)}$ are simply $\mathcal{C}_h^{l(k)}$. The minimum entropy requirement from Section 3 then gives us that for any h which has non-zero probability, $H_\infty(\mathcal{C}_h^{(k)}) = \omega(\log k)$.

Let $h_1, h_2, ..., h_m$ be any sequence of histories which all have non-zero probability under $\mathcal{C}^{(k)}$ and let $f : \{0,1\}^{m(k)} \times D \times \{0,1\}$ be a universal hash function. Let $Y, Z \leftarrow U_{m(k)}$, and $D_i \leftarrow \mathcal{C}_{h_i}^{(k)}$. Let $L(k) = \min_i H_\infty(D_i)$, and note that $L(k) = \omega(\log k)$. Then the "Leftover Hash Lemma" (see, e.g., [12]) implies that

$$|\Pr_{Y,D_i}[(Y, f_Y(D_1), ..., f_Y(D_m)) = (y, b_1, ..., b_n)] - \Pr_{Z,B \leftarrow \{0,1\}^m}[(Z,B) = (y, b_1, ..., b_n)]| \leq m2^{-L(k)/2+1} .$$

from which it is immediate that if we choose $Y \leftarrow U_{m(k)}$ once and publicly, then for all $1 \leq i \leq m$, f_Y will have negligible bias for $\mathcal{C}_{h_i}$ except with negligible probability.

The same approach can be applied in the case that f is a pseudorandom function, since a random function will have neglible bias for $\mathcal{C}_{h_i}^{(k)}$ except with negligible probability, and testing for any specific non-negligible bias can be accomplished by a polynomial time oracle machine. Note that in this scenario, we only need the pseudorandomness of f to prove that it is unbiased, and so it is permissible to publish a single choice of key once and for all.

D IND\$-CPA Public-Key Encryption

We show how to construct IND\$-CPA public-key encryption schemes from a variety of well-established cryptographic assumptions.

Definition. A *trapdoor one-way predicate family* P is a sequence $\{P_k\}_k$, where each P_k is a set of efficiently computable predicates $p : D_p \to \{0,1\}$, along with an algorithm $G(1^k)$ that samples pairs (p, S_p) uniformly from P_k; S_p is an algorithm that, on input b (a bit) samples x uniformly from D_p subject to $p(x) = b$. For a PPT $\mathbf{A}$ running in time $t(k)$, denote the advantage of $\mathbf{A}$ against P by

$$\mathbf{Adv}^{\mathsf{tp}}_P(\mathbf{A}, k) = \Pr_{(p,S_p)\leftarrow G(1^k), x \leftarrow D_p}[\mathbf{A}(x, S_p) = p(x)] .$$

Define the insecurity of P by $\mathbf{InSec}^{\mathsf{tp}}_P(t,k) = \max_{\mathbf{A}\in\mathcal{A}(t)} \left\{\mathbf{Adv}^{\mathsf{tp}}_P(\mathbf{A},k)\right\}$, where $\mathcal{A}(t)$ denotes the set of all adversaries running in time $t(k)$. We say that P is a trapdoor one-way predicate family if for every probabilistic polynomial-time (PPT) $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{tp}}_P(\mathbf{A},k)$ is negligible in k.

IND\$-CPA public-key encryption schemes can be constructed from any primitive which implies trapdoor one-way predicates p with domains D_p satisfying one of the following conditions:

- D_p is computationally or statistically indistinguishable from $\{0,1\}^{poly(k)}$: in this case it follows directly that encrypting the bit b by sampling from $p^{-1}(b)$ yields an IND\$-CPA scheme. The results of Goldreich and Levin imply that such predicates exist if there exist trapdoor one-way permutations on $\{0,1\}^k$, for example.
- D_p has an efficiently recognizable, polynomially dense encoding in $\{0,1\}^{poly(k)}$; in this case, we let $q(\cdot)$ denote the polynomial such that every D_p has density at least $1/q(k)$. Then to encrypt a bit b, we draw $\ell = kq(k)$ samples $d_1, \ldots, d_\ell \leftarrow U_{poly(k)}$; let i be the least i such that $d_i \in D_p$; then transmit $d_1, \ldots, d_{i-1}, p^{-1}(b), d_{i+1}, \ldots, d_\ell$. (This assumption is similar to the requirement for common-domain trapdoor systems used by [6], and all (publicly-known) public-key encryption systems seem to support construction of trapdoor predicates satisfying this condition.)

Stronger assumptions allow construction of more efficient schemes. Here we will construct schemes satisfying IND\$-CPA under the following assumptions: trapdoor one-way permutations on $\{0,1\}^k$ (Section D.1), the RSA assumption (D.2), and the Decisional Diffie-Hellman assumption (D.3). Notice that although both of the latter two assumptions imply the former through standard constructions, the standard constructions exhibit considerable security loss which can be avoided by our direct constructions.

D.1 Efficient Probabilistic Encryption

The following "EPE" encryption scheme is described in [11], and is a generalization of the protocol given by [3]. When used in conjunction with a family of trapdoor one-way permutations on domain $\{0,1\}^k$, it is easy to see that the scheme satisfies IND\$-CPA:

Construction 6. (EPE Encryption Scheme)

Procedure `Encrypt`:
Input: plaintext m, trapdoor OWP π
Sample $x_0, r \leftarrow U_k$
let $l = |m|$
for $i = 1 \ldots l$ do
  set $b_i = x_{i-1} \odot r$
  set $x_i = f(x_{i-1})$
Output: $x_l, r, b \oplus m$

Procedure `Decrypt`:
Input: Ciphertext x, r, c, trapdoor π^{-1}
let $l = |c|$, $x_l = x$
for $i = l \ldots 1$ do
  set $x_{i-1} = \pi^{-1}(x_i)$
  set $b_i = x_{i-1} \odot r$
Output: $c \oplus b$

IND\$-CPA-ness follows by the pseudorandomness of the bit sequence $b_1, \ldots, b_l$ generated by the scheme and the fact that x_l is uniformly distributed in $\{0,1\}^k$.

D.2 RSA-based construction

The RSA function $E_{N,e}(x) = x^e \bmod N$ is a trapdoor one-way permutation family with dense domains, and can be transformed through standard constructions to a trapdoor OWP family on domain $\{0,1\}^k$, but such transformation incurs a heavy security loss. Here we give a direct application of the previous scheme which uses Young and Yung's Probabilistic Bias Removal Method (PBRM) to ensure that x_l from the previous scheme is uniformly distributed on $\{0,1\}^k$ rather than $\mathbb{Z}_N$:

Construction 7. (Bias-corrected RSA-based EPE Encryption Scheme)

Procedure `Encrypt`:
Input: plaintext m; public key N, e
let $k = |N|$, $l = |m|$
repeat:
  Sample $x_0 \leftarrow \mathbb{Z}_N^*$
  for $i = 1 \ldots l$ do
    set $b_i = x_{i-1} \bmod 2$
    set $x_i = x_{i-1}^e \bmod N$
  sample $c \leftarrow U_1$
until $(x_l \leq 2^k - N)$ OR $c = 1$
if $(x_1 \leq 2^k - N)$ and $c = 0$ set $x' = x$
if $(x_1 \leq 2^k - N)$ and $c = 1$ set $x' = 2^k - x$
Output: $x', b \oplus m$

Procedure `Decrypt`:
Input: Ciphertext x', c; private key N, d
let $l = |c|$, $k = |N|$
if $(x' > N)$ set $x_l = x'$
else set $x_l = 2^k - x'$
for $i = l \ldots 1$ do
  set $x_{i-1} = x_i^d \bmod N$
  set $b_i = x_{i-1} \bmod 2$
Output: $c \oplus b$

The IND\$-CPA security of the scheme follows from the correctness of PBRM and the fact that the least-significant bit is a hardcore bit for RSA. Notice that the expected number of repeats in the encryption routine is at most 2.

D.3 DDH-based construction

Let $E_{(\cdot)}(\cdot)$, $D_{(\cdot)}(\cdot)$ denote the encryption and decryption functions of a *private-key* encryption scheme satisfying IND\$-CPA, keyed by κ-bit keys, and let $\kappa \leq k/3$ (private-key IND\$-CPA encryption schemes have appeared in the literature; see, for instance, [13]). Let H_k be a family of pairwise-independent hash functions $h : \{0,1\}^k \to \{0,1\}^\kappa$. We let P be a k-bit prime (so $2^{k-1} < P < 2^k$), and let $P = rQ + 1$ where $(r, Q) = 1$ and Q is also a prime. Let g generate $\mathbb{Z}_P^*$ and $\hat{g} = g^r \bmod P$ generate the unique subgroup of order Q. The security of the following scheme follows from the Decisional Diffie-Hellman assumption, the leftover-hash lemma, and the security of (E, D):

Construction 8. (ElGamal-based random-bits encryption)

Procedure `Encrypt`:
Input: plaintext m; public key $g, \hat{g}^a, P$
Sample $h \leftarrow H_k$
repeat:
  Sample $b \leftarrow \mathbb{Z}_{P-1}$
until $(g^b \bmod P) \leq 2^{k-1}$
set $K = h((\hat{g}^a)^b \bmod P)$
Output: $h, (g^b \bmod P) \bmod 2^{k-1}, E_K(m)$

Procedure `Decrypt`:
Input: Ciphertext h, s, c; private key a, P, Q
let $r = (P-1)/Q$
set $K = h(s^{ra} \bmod P)$
Output: $D_K(c)$

The security proof considers two hybrid encryption schemes: H_1 replaces the value $(\hat{g}^a)^b$ by a random element of the subgroup of order Q, $\hat{g}^c$, and H_2 replaces K by a random draw from $\{0,1\}^\kappa$. Clearly distinguishing H_2 from random bits requires distinguishing some $E_K(m)$ from random bits. The Leftover Hash Lemma gives that the statistical distance between H_2 and H_1 is at most $2^{-\kappa}$. Finally, any q-query distinguisher for H_1 from the output of `Encrypt` with advantage ϵ can be used to solve the DDH problem with advantage at least $\epsilon/2q$, using the same technique from Lemma 2 and a standard hybrid argument.