Public-Key Steganography

Luis von Ahn and Nicholas J. Hopper

Computer Science Dept, Carnegie Mellon University, Pittsburgh PA 15213 USA

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 [7] 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 (a steganographic analogue of Rackoff and Simon's attacker-specific adaptive chosen-ciphertext adversaries from CRYPTO 91 [10]).

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 [11] 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 to a protocol which is provably secure assuming that one-way functions exist [7]. 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.

C. Cachin and J. Camenisch (Eds.): EUROCRYPT 2004, LNCS 3027, pp. 323–341, 2004.

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 (a steganographic analogue of attacker-specific adaptive chosen-ciphertext adversaries [10]).

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 [4], and several papers have since given similar models [8,9,14]. 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 [7] 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 [5], have both previously described ideas for public-key steganography with only heuristic arguments for security. Since our work has been distributed, others have presented ideas for improving the efficiency of our basic scheme [12] and proposing a modification which makes the scheme secure against a more powerful active adversary [2].

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 attacker-specific adaptive chosen-ciphertext adversaries [10]); we show that security against such adversaries can be achieved in the random oracle model. We stress, however, 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 $\mu : \mathbb{N} \rightarrow [0, 1]$ 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 $H_\infty(\mathcal{C}) = \min_{x\in X}\{\log_2(1/\Pr_{\mathcal{C}}[x])\}$. 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.

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 [7] 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 $s \in D^*$.

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 $\mathcal{C}_{A\rightarrow B,h}$ to denote the 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 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 $\mathcal{C}_{P\rightarrow Q}$. In these cases, we assume that among the legitimate parties, only party A has oracle access to marginal channel distributions $\mathcal{C}_{A\rightarrow B,h}$ for every other party B and history h. On the other hand, we will allow the adversary oracle access to marginal channel distributions $\mathcal{C}_{P\rightarrow Q,h}$ 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 $\mathcal{C}_{A\rightarrow B,h}$.

4 Pseudorandom Public-Key Encryption

We will require public-key encryption schemes that are secure in a slightly non-standard model, which we will denote by $\mathsf{IND\$\text{-}CPA}$ in contrast to the more standard $\mathsf{IND\text{-}CPA}$. The main difference is that security against $\mathsf{IND\$\text{-}CPA}$ requires the output of the encryption algorithm to be indistinguishable from uniformly chosen random bits. Let $\mathcal{E} = (G, E, D)$ be a probabilistic public-key encryption scheme, where $E : \mathcal{PK} \times \mathcal{R} \times \mathcal{P} \rightarrow \mathcal{C}$. Consider a game in which an adversary $\mathbf{A}$ is given a public key drawn from $G(1^k)$ and chooses a message $m_{\mathbf{A}}$. Then $\mathbf{A}$ is given either $E_{PK}(m_{\mathbf{A}})$ or a uniformly chosen string of the same length. Let $\mathcal{A}(t, l)$ be the set of adversaries $\mathbf{A}$ which produce a message of length at most $l(k)$ bits and run for at most $t(k)$ time steps. Define the $\mathsf{IND\$\text{-}CPA}$ advantage of $\mathbf{A}$ against $\mathcal{E}$ as

$$\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) = \left| \Pr_{PK}[\mathbf{A}(PK, E_{PK}(m_{\mathbf{A}})) = 1] - \Pr_{PK}[\mathbf{A}(PK, U_{|E_{PK}(m_{\mathbf{A}})|}) = 1] \right|$$

Define the insecurity of $\mathcal{E}$ as $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, l, k) = \max_{\mathbf{A} \in \mathcal{A}(t,l)} \left\{ \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) \right\}$. $\mathcal{E}$ is (t, l, k, ϵ)*-indistinguishable from random bits under chosen plaintext attack* if $\mathbf{InSec}^{\mathsf{cpa}}_{\mathcal{E}}(t, l, k) \leq \epsilon(k)$. $\mathcal{E}$ is called *indistinguishable from random bits under chosen plaintext attack* ($\mathsf{IND\$\text{-}CPA}$) if for every probabilistic polnyomial-time (PPT) $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k)$ is negligible in k. For completeness, we show how to construct $\mathsf{IND\$\text{-}CPA}$ public-key encryption schemes from the RSA and Decisional Diffie-Hellman assumptions. We omit detailed proofs of security for the constructions below, as they are standard modifications to existing schemes. In the full version of this paper, we show that much more general assumptions suffice for $\mathsf{IND\$\text{-}CPA}$ security.

4.1 RSA-based Construction

The RSA function $E_{N,e}(x) = x^e \bmod N$ is believed to be a trapdoor one-way permutation family. The following construction uses Young and Yung's Probabilistic Bias Removal Method (PBRM) [13] to remove the bias incurred by selecting an element from $\mathbb{Z}^*_N$ rather than U_k.

Construction 1. (RSA-based Pseudorandom 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 $x_i = x^e_{i-1} \bmod N$
  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: $x', c$; $(N, d)$
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 $x_{i-1} = x^d_i \bmod N$
    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.

4.2 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, [7]). Let $\mathcal{H}_k$ be a family of pairwise-independent hash functions $H : \{0,1\}^k \rightarrow \{0,1\}^\kappa$. 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 2. (ElGamal-based random-bits encryption)

```
**Procedure** Encrypt:
**Input:** $m \in \{0,1\}^*$; $(g, \hat{g}^a, P)$
Sample $H \leftarrow \mathcal{H}_k$
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:** $H, g^b, E_K(m)$
```

```
**Procedure** Decrypt:
**Input:** $(H, s, c)$; 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 distinguisher $\mathbf{A}$ for H_1 from the output of `Encrypt` with advantage ϵ can be used to solve the DDH problem with advantage at least $\epsilon/2$, by transforming $\hat{g}^b$ to $B = (\hat{g}^b)^{\hat{r}} g^{\beta Q}$, where $\hat{r}$ is the least integer such that $r\hat{r} = 1 \bmod Q$ and $\beta \leftarrow \mathbb{Z}_r$, outputting 0 if $B > 2^{k-1}$, and otherwise drawing $H \leftarrow \mathcal{H}_k$ and running $\mathbf{A}$ on $\hat{g}^a, H||B||E_{H(\hat{g}^c)}(m)$.

5 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. 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*. We also 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 $\mathcal{C}_h$.

5.1 Steganographic Secrecy against Chosen Hiddentext Attack

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. (Note that a chosen hiddentext attack is the steganographic equivalent of a chosen plaintext attack against an encryption scheme.)

We will model a warden attacking a stegosystem as an efficient oracle machine which plays the following game. First, a pair (PK, SK) is drawn from $SG(1^k)$ and W is allowed to access PK and $\mathcal{C}_h$ for any history h, to produce a message m_W and a history h_W. Then W is given access to a sequence $s \in D^*$, which is drawn either from $SE(PK, m_W, h_W)$ or $\mathcal{C}_h^{|s|}$, and continues to access $\mathcal{C}_h$ for any history h (This oracle allows W to learn the *covertext* distribution on all communications). Finally W outputs a bit which is a "guess" about whether s is a stegotext or a covertext.

We define the *CHA advantage of W against stegosystem S over channel $\mathcal{C}$* by

$$\mathbf{Adv}^{\mathsf{cha}}_{S,\mathcal{C}}(W,k) = \left| \Pr[W^{\mathcal{C}}(PK, s) = 1] - \Pr[W^{\mathcal{C}}(PK, \mathcal{C}_{h_W}^{|s|}) = 1] \right| ,$$

where the probabilities are taken over $(PK, SK) \leftarrow SG(1^k)$, $(m_W, h_W) \leftarrow W^{\mathcal{C}}(PK)$, $s \leftarrow SE(PK, m_W, h_W)$, the random bits of W and the responses of the oracle $\mathcal{C}$. Define *the insecurity of S over channel $\mathcal{C}$* by $\mathbf{InSec}^{\mathsf{cha}}_{S,\mathcal{C}}(t,l,k) = \max_{W \in \mathcal{W}(t,l)} \left\{ \mathbf{Adv}^{\mathsf{cha}}_{S,\mathcal{C}}(W,k) \right\}$, where $\mathcal{W}(t,l)$ denotes the set of all adversaries which choose a message of length at most $l(k)$ bits, and run in time at most $t(k)$. For ease of notation, we will denote $W^{\mathcal{C}}$ by W for the remainder of the paper.

Thus, S is secure against chosen-hiddentext attacks if the warden, even given Bob's public key, cannot tell whether Alice's message to Bob encodes any hiddentext at all, even one of his choice.

5.2 Chosen-Stegotext Attacks

In a Chosen-Stegotext Attack, Ward is monitoring the communications from Alice to Bob, trying to detect any hint that the pair are using steganography. In particular, he not only has some partial knowledge of the messages Alice might be encoding, he can pretend to be Alice, sending messages (presumably encoding hiddentexts) to Bob and observing Bob's reaction to see if Bob is trying to decode them. Ward can even make small changes to Alice's messages to see how Bob reacts to them. If Bob is decoding Alice's messages, he will output hiddentexts, whereas otherwise he will have no special reaction, which we signify by the output $\perp$. In order to protect against this attack, there must be something Alice knows that Ward doesn't. So we extend the definition of a stegosystem to include a fourth algorithm $SSG(1^k)$ which outputs a pair (KV, KS). If Alice wants to send messages that are protected in this way, she chooses $(KV, KS) \leftarrow SSG(1^k)$ and publishes KV in the same way that Bob would publish PK. We modify the encoding algorithm to take KS as an additional argument, and the decoding algorithm to require KV. (So in a system designed to resist this attack, Bob must know Alice's public value KV to decode her messages.)

We formally model a chosen-stegotext attacker as an oracle PPT W which plays the following game. First, draw $(PK, SK) \leftarrow SG(1^k)$, and $(KV, KS) \leftarrow SSG(1^k)$. Then W is given (PK, KV) and accesses an oracle, which is either ST (for StegoText) or CT (for CoverText), and which respond to queries using these programs: (where $\phi = \{\}$ initially):

```
$ST(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if(b = $\texttt{enc}$) then:
  Sample $s \leftarrow SE(PK, KS, m, h)$
  Set $\phi = \phi \cup \{(s, h)\}$; return $s$
else if $(m, h) \in \phi$ return “”
else return $SD(SK, KV, m, h)$
```

```
$CT(b \in \{\texttt{enc}, \texttt{dec}\}, m, h)$
if(b = $\texttt{enc}$) then:
  Sample $s \leftarrow \mathcal{C}_h^{|SE(PK,KS,m,h)|}$
  Set $\phi = \phi \cup \{(s, h)\}$; return $s$
else if $(m, h) \in \phi$ return “”
else return $\perp$
```

Suppose W runs in time t, and makes q_e queries of the form $(\texttt{enc}, *, *)$, totalling l bits, and q_d queries of the form $(\texttt{dec}, *, *)$. We define W's CSA advantage against S for channel $\mathcal{C}$ by

$$\mathbf{Adv}^{\mathsf{csa}}_{S,\mathcal{C}}(W, k) = \left|\Pr[W^{ST}(PK, KV) = 1] - \Pr[W^{CT}(PK, KV) = 1]\right| \ ,$$

and the insecurity by $\mathbf{InSec}^{\mathsf{csa}}_{S,\mathcal{C}}(t, q_e, q_d, l, k) = \max_{w \in \mathcal{W}(t, q_e, q_d, l)} \left\{\mathbf{Adv}^{\mathsf{csa}}_{S,\mathcal{C}}(W, k)\right\}$.

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. The use of a secret by the encryptor makes this security condition roughly analogous to the notion of "attacker-specific" adaptive chosen-ciphertext attack from [10], in which an attacker who tries to access the decryption of a message he didn't send is given the response $\perp$. This "attacker-specific" notion of security is stronger than CCA-1 security but weaker than the now-standard CCA-2 security [10]. Note, however, that chosen-stegotext security protects *both* the sender and the receiver.

5.3 Relationships among Notions

We formalize the notions of steganographic secrecy as follows.

Definition 2. A stegosystem S is called steganographically secret for channel $\mathcal{C}$ against attack model atk ($\mathsf{SS\text{-}ATK}$) if for every PPT $\mathbf{A}$, $\mathbf{Adv}^{\mathsf{atk}}_{S,\mathcal{C}}(\mathbf{A},k)$ is negligible in k.

A natural question is: what are the relationships between these security notions and the standard notions from public-key cryptography? In this section we give the key relationships between these notions.

$\mathsf{SS\text{-}CHA}$ is strictly stronger than $\mathsf{IND\text{-}CPA}$. By a standard argument based on the triangle inequality, if A can distinguish $SE(m_0)$ from $SE(m_1)$ with advantage ϵ, he must be able to distinguish one of these from $\mathcal{C}_h$ with advantage at least $\epsilon/2$. Thus every $\mathsf{SS\text{-}CHA}$ secure stegosystem must also be $\mathsf{IND\text{-}CPA}$ secure. On the other hand, let S be any $\mathsf{IND\text{-}CPA}$ secure cryptosystem. Then S' which prepends a known, fixed sequence of documents $m \in D^k$ to the output of S is still $\mathsf{IND\text{-}CPA}$ secure but has an $\mathsf{SS\text{-}CHA}$ distinguisher with advantage $1 - o(1)$ for any L-informative channel.

$\mathsf{SS\text{-}CSA}$ is strictly stronger than $\mathsf{SS\text{-}CHA}$. Suppose that we take a $\mathsf{SS\text{-}CSA}$-secure stegosystem $S = (SG, SSG, SE, SD)$ and define $SE'(PK, m, h)$ to draw a random $(KV, KS) \leftarrow SSG(1^k)$ and return $SE(PK, KS, m, h)$. Then any CHA warden against SE' is also a single-query CSA warden against S. (However, whether there is a corresponding modification SD' so that S' is sound may be dependent on the construction; such modification is possible for our construction.) On the other hand, $\mathsf{SS\text{-}CSA}$ is strictly stronger than $\mathsf{SS\text{-}CHA}$: if (SG, SE, SD) is $\mathsf{SS\text{-}CHA}$ secure, then so is $S' = (SG, SE', SD')$ where $SE'(m, h)$ draws $s \leftarrow SE(m, h)$ and $s' \leftarrow \mathcal{C}_{(h,s)}$, and returns (s, s'), while $SD'((s, s'), h)$ returns $SD(s, h)$. But S' is trivially vulnerable to a chosen-stegotext attack with advantage 1: query $(\mathtt{enc}, m, h)$ to get (s, s'), draw $s'' \leftarrow \mathcal{C}_{(h,s)}$ and query $(\mathtt{dec}, (s, s''), h)$. If the result is not $\perp$, return 1, otherwise return 0.

6 Constructions

Most of our protocols build on the following construction, a generalization of Construction 2 in [7] and similar to a protocol given by Cachin [4]. Let $f : D \rightarrow \{0, 1\}$ 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 $\mathcal{C}_h^l$:

Construction 3. (Basic encoding/decoding routines)

```
Procedure Basic_Encode:
Input: $c_1, \ldots, c_l \in \{0,1\}^l$, $h \in D^*$, $k$
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 B, 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 $h_1, \ldots, h_n$ all satisfying $\Pr_{\mathcal{C}}[h_i] \neq 0$, the bias $\left|\Pr[f(\mathcal{C}_{h_i}^{(k)}) = 1] - \frac{1}{2}\right|$ is negligible in k for all $1 \leq i \leq n$.*

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

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

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: it is possible to have $f(s_i) \neq c_i$. 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 4. (Chosen Hiddentext Security)

```
Procedure CHA_Encode:
Input: key $PK$, $m \in \{0,1\}^*$, $h \in D^*$
Let $c = E_{PK}(m)$
Output: Basic_Encode$(c, h, k)$
```

```
Procedure CHA_Decode:
Input: secret key $SK$, $s \in D^l$
Let $c = $ Basic_Decode$(s)$
Output: $D_{SK}(c)$
```

Lemma 1. *Assume f is ϵ-biased on $\mathcal{C}_h$ for all h. For any warden $W \in \mathcal{W}(t, l)$, there is an* IND\$-CPA *adversary* $\mathbf{A}$ *with* $\mathbf{Adv}^{\mathsf{cha}}_{\mathtt{CHA},\mathcal{C}}(W, k) \leq \mathbf{Adv}^{\mathsf{cpa}}_{\mathcal{E}}(\mathbf{A}, k) + \ell(l)\epsilon$. *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* $\mathbf{A}$ *submits a plaintext of length at most* l.

Proof. We are given a program W for distinguishing between the output of Construction 4 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 $E_{PK}(m_W)$ from $U_{\ell(l)}$. $\mathbf{A}(PK)$ simply runs $W(PK)$, to get (m_W, h_W) and returns m_W for m_A. Then $\mathbf{A}(PK, c)$ uses the oracle $\mathcal{C}_h$ to compute $s = \mathtt{Basic\_Encode}(c, h_W, k)$, and returns the output of $W(PK, s)$. Consider the cases for $\mathbf{A}$'s input. If $c \leftarrow E_{PK}(m_W)$, then $s \leftarrow \mathtt{CHA\_Encode}(PK, m_W, h_W)$; so we have $\Pr[\mathbf{A}(PK, E_{PK}(m_A)) = 1] = \Pr[W(PK, SE(PK, m_W, h_W)) = 1]$. If $c \leftarrow U_\ell$, then s is distributed identically to $\mathtt{Basic\_Encode}(U_\ell, h_W, k)$; thus $|\Pr[\mathbf{A}(PK, U_\ell) = 1] - \Pr[W(PK, C^\ell_{h_W}) = 1]| \leq \ell\epsilon$ by Proposition 2.

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

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

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

6.2 Chosen-Stegotext Security

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. 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 [3], 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 UEncode:
Input: $c \in \{0,1\}^l$, $r$, $h$
for $i = 1 \ldots l$ do
  Let $j = 0$
  repeat:
    set $r_j = G(h, r, c, j)$
    set $s_i = \mathtt{C}(h; r_j)$
    increment $j$
  until $f(s_i) = c_i \vee (j > k)$
  set $h = (h, s_i)$
Output: $s_1, s_2, \ldots, s_l$
```

```
Procedure CSA_Encode:
Input: $m$, $h$, $\pi_A^{-1}$, $\pi_B$
Choose $r \leftarrow U_k$
Let $\sigma = \pi_A^{-1}(F(r, m, h))$
Let $e = H_1(r) \oplus (m, \sigma)$
Let $\tau = H_2(r, m, h)$
Let $y = \pi_B(r)$
Let $c = y||e||\tau$
Output: $\mathtt{UEncode}^G(c, r, h)$
```

```
Procedure CSA_Decode:
Input: $s$, $h$, $\pi_A$, $\pi_B^{-1}$
Let $c = \mathtt{Basic\_Decode}(s)$
Parse $c$ as $y||e||\tau$.
Set $r = \pi_B^{-1}(y)$.
Let $(m, \sigma) = e \oplus H_1(r)$
If $s \neq \mathtt{UEncode}^G(c, r, h) \vee$
  $\tau \neq H_2(r, m, h) \vee$
  $\pi_A(\sigma) \neq F(r, m, h)$
  return $\perp$
Output: $m$
```

Theorem 2. *If f is ϵ-biased for $\mathcal{C}$, then*

$$\mathbf{InSec}^{\mathsf{csa}}_{\mathsf{CSA},\mathcal{C}}(t,\boldsymbol{q},l,k) \leq (2q_e+q_F)\mathbf{InSec}^{\mathsf{ow}}_{\pi}(t',k)+(l+3q_ek)\epsilon+(q_e^2+2q_d)/2^k \ ,$$

where $t' \leq t+(q_G+q_F+q_H)(q_e+q_d)T_\pi + k(l+3q_ek)T_{\mathsf{C}}$, T_π is the time to evaluate members of π, and T_{C} is the running time of 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 A for details.

7 Steganographic Key Exchange

Consider the original motivating scenario: Alice and Bob are prisoners, in an environment controlled by Ward, who wishes to prevent them from exchanging messages he can't read. Then the best strategy for Ward, once he has read the preceding sections, is to ban Alice and Bob from publishing public keys. In this case, 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 $\mathcal{T}_{r_A,r_B}$ (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.

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

$$\mathbf{Adv}^{\mathsf{ske}}_{S_{KE},\mathcal{C}}(W,k) = |\Pr[W(\mathcal{T}_{r_A,r_B}, S_{KE}(r_A,r_B)) = 1] - \Pr[W((\sigma_A,\sigma_B),K) = 1]|$$

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 $\mathcal{C}$ with security parameter k* is given by $\mathbf{InSec}^{\mathsf{ske}}_{S_{KE},\mathcal{C}}(t,k) = \max_{W\in\mathcal{W}(t)}\left\{\mathbf{Adv}^{\mathsf{ske}}_{S_{KE},\mathcal{C}}(W,k)\right\}$.

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.

Construction. 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 6. (Steganographic Key Exchange)

```
Procedure SKE_Encode$_A$:
Input: $(P, Q, h, g)$
repeat:
  sample $a \leftarrow U(\mathbb{Z}_{P-1})$
until $g^a \mod P < 2^{k-1}$
Let $c_a = g^a \mod 2^{k-1}$
Output: Basic_Encode$(c_a, h, k)$
```

```
Procedure SKE_Decode$_A$:
Input: $s \in D^l$, exponent $a$
Let $c_b$ = Basic_Decode$(s)$
Output: $c_b^{ra} \mod P = \hat{g}^{ab}$
```

(`SKE_Encode`$_B$ and `SKE_Decode`$_\mathtt{B}$ are analogous)

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* $\mathbf{Adv}^{\mathsf{ddh}}_{\hat{g},P,Q}(\mathbf{A}) \ge \frac{1}{4}\mathbf{Adv}^{\mathsf{ske}}_{\mathsf{SKE}}(W,k) - \epsilon|P|$. *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 $W(\texttt{Basic\_Encode}(c_a, h_A, k), \texttt{Basic\_Encode}(c_b, h_B, k), \hat{g}^c)$, 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 $\mathcal{C}_{A\rightarrow B,h_A}$ and $\mathcal{C}_{B\rightarrow A,h_B}$ for all h_A, h_B, then*

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

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 occurred 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 4, 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. In contrast to the private-key results of [7], we are 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.

Acknowledgements. The authors wish to thank Manuel Blum for his constant support and encouragement. We thank Christian Cachin for his helpful comments on our submitted draft, as well as Adam Bender, Andrew Bortz, Tri Van Le, Ke Yang, and several anonymous CRYPTO and Eurocrypt referees. This material is based upon work partially supported by the National Science Foundation under Grants CCR-0122581 and CCR-0058982 (The Aladdin Center). This work was also partially supported by the Army Research Office (ARO) and the Center for Computer and Communications Security (C3S) at Carnegie Mellon University. The first author is partially supported by a Microsoft Research Fellowship and the second author has been partially supported by a NSF Graduate Research Fellowship and a Siebel Scholarship.

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. M. Backes and C. Cachin. Public-Key Steganography with Active Attacks. Cryptology ePrint Archive, Report 2003/231, November 6, 2003. Available electronically: `http://eprint.iacr.org/2003/231`.
3. M. Bellare and P. Rogaway. Random Oracles are Practical. *Computer and Communications Security: Proceedings of ACM CCS 93*, pages 62–73, 1993.
4. C. Cachin. An Information-Theoretic Model for Steganography. *Proceedings of Second International Information Hiding Workshop*, Springer LNCS 1525, pages 306-318, 1998.
5. S. Craver. On Public-key Steganography in the Presence of an Active Warden. *Proceedings of Second International Information Hiding Workshop*, Springer LNCS 1525, pages 355-368, 1998.
6. 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.
7. N. J. Hopper, J. Langford, and L. Von Ahn. Provably Secure Steganography. *Advances in Cryptology: CRYPTO 2002*, Springer LNCS 2442, pages 77-92, 2002.
8. T. Mittelholzer. An Information-Theoretic Approach to Steganography and Watermarking. *Proceedings of the Third International Information Hiding Workshop*, Springer LNCS 1768, pages 1-16, 2000.
9. J. A. O'Sullivan, P. Moulin, and J. M. Ettinger. Information-theoretic analysis of Steganography. *Proceedings ISIT 98*, 1998.
10. C. Rackoff and D. Simon. Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. *Advances in Cryptology: CRYPTO 91*, Springer LNCS 576, pages 433-444, 1992.

11. G. J. Simmons. The Prisoner's Problem and the Subliminal Channel. *Advances in Cryptology: CRYPTO 83*, pages 51-67, 1983.
12. 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`.
13. A. Young and M. Yung. Kleptography: Using Cryptography against Cryptography. *Advances in Cryptology: Eurocrypt 97*, Springer LNCS 1233, pages 62-74, 1997.
14. 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-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 `CSA_Encode` but with calls to $\texttt{UEncode}^G$ replaced by calls to `Basic_Encode`.
3. $\mathsf{P2}(b,m,h)$ responds to `dec` queries as in $\mathsf{P1}$, and responds to `enc` queries using `CSA_Encode`.
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 $\mathbf{Adv}^{\mathsf{csa}}_{\mathtt{CSA},\mathcal{C}}(W,k) \leq |\Pr[W^{P0}=1] - \Pr[W^{P1}=1]| + |\Pr[W^{P1}=1] - \Pr[W^{P2}=1]| + |\Pr[W^{P2}=1] - \Pr[W^{P3}=1]|$ (for ease of notation, we omit the arguments π_A, π_B to W). Hence, we can bound the advantage of W by the sum of its advantages in distinguishing the successive hybrids. For hybrids P, Q we let $\mathbf{Adv}^{\mathsf{P},\mathsf{Q}}(W,k) = |\Pr[W^{\mathsf{P}}=1] - \Pr[W^{\mathsf{Q}}=1]|$.

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

Proof. Assume WLOG that $\Pr[W^{P1}=1] > \Pr[W^{P0}=1]$. 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 `CSA_Encode`, and let $E \equiv E_r \wedge E_q$.

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

because if r never repeats and W never queries $H_1(r)$ or $H_2(r, *, *)$ for some r used by CSA_Encode, 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 CSA_Encode but with calls to UEncode^G replaced by calls to Basic_Encode. On the i-th query respond with $s =$ 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 $\Pr[W^{P2} = 1] > \Pr[W^{P1} = 1]$. Denote by E_r the event that, when answering queries for W, the random value r of CSA_Encode never repeats, and by E_q the event that W never queries $G(*, r, \pi_B(r)||*, *)$ for some r used by CSA_Encode, and let $E \equiv E_r \wedge E_q$. Then:

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

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:

- enc queries are answered as follows: on query $j \neq i$, respond using CSA_Encode. On the i-th query respond with $s =$ 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 $\phi = \phi \cup \{(s, h)\}$.

- 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:

- enc queries are answered using CSA_Encode 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.
- dec queries are answered using CSA_Decode, 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 CSA_Decode. 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 some $F(r,m,h)$, we have that: $\mathbf{Adv}^{\mathsf{P2,P3}}(W,k) \leq q_F \mathbf{InSec}^{\mathsf{ow}}_{\Pi}(t',k) + q_d/2^{k-1} + q_e/2^k$.

B Negligibly Biased Functions for Any Channel

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 $Y, Z \leftarrow U_{m(k)}, B \leftarrow U_m$, 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., [6]) implies that

$$\varDelta(\langle Y, f_Y(D_1), ..., f_Y(D_m)\rangle, \langle Y, B\rangle) \leq m2^{-L(k)/2+1} \ ,$$

where $\varDelta(X,Y) = \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$ is the statistical distance, 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.