Public-Key Steganography with Active Attacks

Michael Backes Christian Cachin

IBM Research
Zurich Research Laboratory
CH-8803 Rüschlikon, Switzerland
{mbc,cca}@zurich.ibm.com

February 10, 2004

Abstract

A complexity-theoretic model for public-key steganography with active attacks is introduced. The notion of *steganographic security against adaptive chosen-covertext attacks (SS-CCA)* and a relaxation called *steganographic security against replayable adaptive chosen-covertext attacks (SS-RCCA)* are formalized. These notions are closely related with *CCA-security* and *RCCA-security* for public-key cryptosystems. In particular, it is shown that any SS-(R)CCA stegosystem is an (R)CCA-secure public-key cryptosystem and that an SS-RCCA stegosystem can be realized from any RCCA-secure public-key cryptosystem with pseudorandom ciphertexts.

1 Introduction

Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages. As the goal of steganography is to hide the *presence* of a message, it can be seen as the complement of cryptography, whose goal is to hide the *content* of a message.

Consider two parties linked by a public communications channel which is under the control of an adversary. The parties are allowed to exchange messages as long as they are not adding a hidden meaning to their conversation. A genuine communication message is called *covertext*; but if the sender of a message has embedded hidden information in a message, it is called *stegotext*. The adversary, who also knows the distribution of the covertext, tries to detect whether a given message is covertext or stegotext.

Steganography has a long history as surveyed by Anderson and Petitcolas [2], but formal models for steganography have only recently been introduced. Several information-theoretic formalizations [4, 22, 13] and one complexity-theoretic model [11] have addressed *private-key* steganography, where the participants share a common secret key. These models are all limited to a passive adversary, however, who can only read messages on the channel.

In this paper, we introduce a complexity-theoretic model for public-key steganography with active attacks, where the participants a priori do not need shared secret information and the adversary may write to the channel and mount a so-called *adaptive chosen-covertext attack*. This attack seems to be the most general attack conceivable against a public-key stegosystem. It allows the adversary to send an arbitrary sequence of adaptively chosen covertext messages to a receiver and to learn the interpretation of every message, i.e., if the receiver considers a message to be covertext or stegotext, plus the decoding of the embedded message in the latter case. (Note that here and in the sequel, a message on the channel is also called a "covertext" whenever we do not want to distinguish stegotext and covertext in the proper sense.)

We do not address denial-of-service attacks in this work, where the adversary tries to disrupt the hidden communication among the participants. Although they also qualify as "active" attacks and are very important in practice, we think that protection against them can be addressed orthogonally to the methods presented here.

Our model is based on the intuition that a public-key stegosystem essentially is a public-key cryptosystem with the additional requirement that its output conforms to a given covertext distribution. As in the formalization of private-key steganography [4, 11, 15], the covertext distribution is publicly known and accessible only through an oracle that samples the distribution. We introduce the notions of *steganographic security against adaptive chosen-covertext attacks (SS-CCA)* and *steganographic security against replayable adaptive chosen-covertext attacks (SS-RCCA)* and show that they are closely linked to the analogous notions for public-key cryptosystems, called *security against adaptive chosen-ciphertext attacks* (or *CCA-security*) [14] and *security against replayable adaptive chosen-ciphertext attacks* [5] (or *RCCA-security*), respectively. In particular, we show that stegosystems are related to public-key cryptosystems in the following ways:

Theorem 1 (informal statement). *Any SS-(R)CCA stegosystem is an (R)CCA-secure public-key cryptosystem.*

Theorem 2 (informal statement). *An SS-RCCA stegosystem can be constructed from any RCCA-secure public-key cryptosystem whose ciphertexts are pseudorandom (i.e., computationally indistinguishable from a random bit string).*

A corollary of Theorem 2 is that SS-RCCA stegosystems exist in the standard model and in the random oracle model. The stegosystem constructed in the proof of Theorem 2 embeds more hidden bits per stegotext than any previous system. It is not known if a result analogous to Theorem 2 holds for CCA-security; finding an SS-CCA stegosystem remains an interesting open problem.

Our model for public-key steganography is introduced in Section 2, where also the relation to previous work is discussed. Section 3 recalls the definitions of CCA- and RCCA-security for public-key cryptosystems, states our results formally, and presents the proof of Theorem 1. Section 4 gives the construction of an SS-RCCA stegosystem and proves Theorem 2.

2 Definitions

2.1 Notation

A function $f\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is called *negligible* if for every constant $c \geq 0$ there exists $k_c \in \mathbb{N}$ such that $f(k) < \frac{1}{k^c}$ for all $k > k_c$. A (randomized) algorithm is called *efficient* if its running time is bounded by a polynomial except with negligible probability (over the coin tosses of the algorithm).

Let $x \leftarrow y$ denote the algorithm that assigns a value y to x. If $\mathsf{A}(\cdot)$ is a (randomized) algorithm, the notation $x \leftarrow \mathsf{A}(y)$ denotes the algorithm that assigns to x a randomly selected value according to the probability distribution induced by $\mathsf{A}(\cdot)$ with input y over the set of its outputs.

If $\mathcal{S}$ is a probability distribution, then the notation $x \stackrel{R}{\leftarrow} \mathcal{S}$ denotes the algorithm which assigns to x an element randomly selected according to $\mathcal{S}$. If S is a finite set, then the notation $x \stackrel{R}{\leftarrow} S$ denotes the algorithm which assigns to x an element selected at random from S with uniform distribution over S.

If $p(\cdot, \cdot, \cdots)$ is a predicate, the notation

$$\Pr[x \stackrel{R}{\leftarrow} S; y \stackrel{R}{\leftarrow} T; \cdots : p(x, y, \cdots)]$$

denotes the probability that $p(x, y, \cdots)$ will be true after the ordered execution of the algorithms $x \stackrel{R}{\leftarrow} S$, $y \stackrel{R}{\leftarrow} T, \cdots$. If X is a (randomized) algorithm, a distribution, or a set, then $\Pr_X[x]$ is short for $\Pr_{x \stackrel{R}{\leftarrow} X}[x]$, which is short for $\Pr[s \stackrel{R}{\leftarrow} X : s = x]$.

The *statistical distance* between two distributions $\mathcal{X}$ and $\mathcal{Y}$ over the same set X is defined as $\|\mathcal{X} - \mathcal{Y}\| = \max_{X_0 \subseteq X} \left|\sum_{x \in X_0} \Pr_{\mathcal{X}}(x) - \Pr_{\mathcal{Y}}(x)\right|$. The *min-entropy* of a distribution $\mathcal{X}$ over an alphabet X is defined as $H_\infty(\mathcal{X}) = -\log \max_{x \in X} \Pr_{\mathcal{X}}[x]$. (All logarithms are to the base 2.)

2.2 Public-key Stegosystems

We define a public-key stegosystem as a triple of algorithms for key generation, message encoding, and message decoding, respectively. The notion corresponds to a public-key cryptosystem in which the ciphertext should conform to a target covertext distribution.

For the scope of this work, the covertext is modeled by a distribution $\mathcal{C}$ over a given set C. The distribution is only available via an oracle; it samples $\mathcal{C}$ upon request, with each sample being independent. In other words, it outputs a sequence of independent and identically distributed covertexts. W.l.o.g., $\Pr_{\mathcal{C}}[c] > 0$ for all $c \in C$.

The restriction to independent repetitions is made here only to simplify the notation and to focus on the contribution of this work. All our definitions and results can be extended in the canonical way to the very general model of a covertext *channel* as introduced by Hopper et al. [11]. They model a channel as an unbounded sequence of values drawn from a set C whose distribution may depend in arbitrary ways on past outputs; access to the channel is given only by an oracle that samples from the channel.

Such a channel underlies only one restriction: The sampling oracle must allow random access to the channel distribution, i.e., the oracle can be queried with an arbitrary prefix of a possible channel output and will return the next symbol according to the channel distribution. In other words, the channel sampler cannot only be rewound to an earlier state of its execution but also restarted from a given state. (Hence it may be difficult to use an email conversation among humans for a covertext channel since that cannot easily be rewound.)

The sampling oracle for the covertext distribution is available to all users and to the adversary. In order to avoid technical complications, assume w.l.o.g. that the sampling oracle is implemented by a probabilistic polynomial-time algorithm and therefore does not help an adversary beyond its own capabilities (for example, with solving a computationally hard problem).

Definition 1. [Public-Key Stegosystem] Let $\mathcal{C}$ be a distribution on a set C of *covertexts*. A *public-key stegosystem* is a triple of probabilistic polynomial-time algorithms $(\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ with the following properties.

- The *key generation algorithm* SK takes as input the security parameter k and outputs a pair of bit strings $(\mathit{spk}, \mathit{ssk})$, called the *[stego] public key* and the *[stego] secret key*.
- The *steganographic encoding algorithm* SE takes as inputs the security parameter k, a public key *spk* and a *message* $m \in \{0,1\}^l$ and outputs a *covertext* $c \in C$. The plaintext m is often called the *embedded message*.
- The *steganographic decoding algorithm* SD takes as inputs the security parameter k, a secret key *ssk*, and a covertext $c \in C$ and outputs either a message $m \in \{0,1\}^l$ or a special symbol $\perp$. An output value of $\perp$ indicates a decoding error, for example, when SD has determined that no message is embedded in c.

We require that for all $(\mathit{spk}, \mathit{ssk})$ output by $\mathsf{SK}(1^k)$ and for all $m \in \{0,1\}^l$, the probability that $\mathsf{SD}(1^k, \mathit{ssk}, \mathsf{SE}(1^k, \mathit{spk}, m)) \neq m$ is negligible in k.

Note that except for the presence of the covertext distribution, this definition is equivalent to that of a public-key cryptosystem. Although all algorithms have oracle access to $\mathcal{C}$, only SE needs it in the stegosystems considered in this paper. For ease of notation, the security parameter will be omitted henceforth.

The probability that the decoding algorithm outputs the correct embedded message is referred to as the *reliability* of the stegosystem. Although one might also allow a non-negligible decoding error in the definition of a stegosystem (as done in previous work [11]), we require that the decoding error probability is negligible in order to maintain the analogy between a stegosystem and a cryptosystem.

Security definition. Coming up with the "right" security definition for a cryptographic primitive has always been a challenging task because the sufficiency of a security property cannot be demonstrated by running the cryptosystem. Only its insufficiency can be shown by pointing out a specific attack, but finding an attack is usually hard. Often, security definitions had to be strengthened when a primitive was used as part of a larger system. Probably the most typical example is the security of public-key cryptosystems: the original notion of semantic security [10], which considers only a passive or eavesdropping adversary, was later augmented to security against adaptive chosen-ciphertext attacks or non-malleability, which allows also for active attacks [14, 9, 3].

We introduce here the notion of *steganographic security against adaptive chosen-covertext attacks*, abbreviated *SS-CCA*, and its slightly relaxed variant *steganographic security against replayable chosen-covertext attacks*, abbreviated *SS-RCCA*. Both notions are based on the intuition that a stegosystem is essentially a cryptosystem with a prescribed ciphertext distribution.

SS-CCA and SS-RCCA are defined by the following experiment. Let an arbitrary distribution $\mathcal{C}$ on a set C be given and consider a (stego-)adversary, defined by two arbitrary probabilistic polynomial-time algorithms SA_1 and SA_2. The experiment consists of five stages where both notions only differ in the fourth stage.

Key generation: A key pair (spk, ssk) is generated by the key generation algorithm SK.

First decoding stage: Algorithm SA_1 is run with the public key *spk* as input and has access to the sampling oracle for $\mathcal{C}$ and to a decoding oracle SO_1. The decoding oracle knows the secret key *ssk*. Whenever it receives a covertext c, it runs $\mathsf{SD}(ssk, c)$ and returns the result to SA_1.

When SA_1 finishes its execution, it outputs a tuple (m^*, s), where $m^* \in \{0,1\}^l$ is a message and s is some additional information which the algorithm wants to preserve.

Challenge: A bit b is chosen at random and a *challenge covertext* c^* is determined depending on it: If $b = 0$ then $c^* \leftarrow \mathsf{SE}(spk, m^*)$ else $c^* \stackrel{R}{\leftarrow} \mathcal{C}$. c^* is given to algorithm SA_2, who should guess the value of b, i.e., determine whether the message m^* has been embedded in c^* or whether c^* has simply been chosen according to $\mathcal{C}$.

Second decoding stage: SA_2 is run on input m^*, c^*, and s, i.e., it knows the message which is potentially embedded, the challenge covertext, and the state provided by SA_1.

For SS-CCA, SA_2 may access a decoding oracle SO_2^{cca}, which is analogous to SO_1 except that upon receiving query c^*, oracle SO_2^{cca} returns `not-allowed`.

For SS-RCCA, SA_2 has access to a decoding oracle SO_2^{rcca}, which is identical to SO_2^{cca} except that SO_2^{rcca} also knows m^* and does not allow certain additional queries to be asked. In particular, upon receiving query c, oracle SO_2^{rcca} computes $m \leftarrow \mathsf{SD}(ssk, c)$, checks if $m \in \{m^*, \perp\}$ and returns `not-allowed` if yes; otherwise, it returns m.

Guessing stage: When SA_2 finishes its execution, it outputs a bit b'.

The stego-adversary succeeds in distinguishing stegotext from covertext if $b' = b$ in the above experiment. We require that for a secure stegosystem, no efficient adversary can distinguish stegotext from covertext except with negligible probability over random guessing.

Definition 2. [Steganographic Security against Active Attacks] Let $\mathcal{C}$ be a distribution on a covertext set C and let $\Sigma = (\mathsf{SK}, \mathsf{SE}, \mathsf{SD})$ be a stegosystem. We say that Σ is *steganographically secure against adaptive chosen-covertext attacks (SS-CCA)* with respect to $\mathcal{C}$ if for all probabilistic polynomial-time adversaries (SA_1, SA_2), there exists a negligible function ϵ such that

$$\Pr\Big[(spk, ssk) \leftarrow \mathsf{SK};\ (m^*, s) \leftarrow SA_1^{SO_1}(spk);\ b \stackrel{R}{\leftarrow} \{0,1\};$$
$$\textbf{if } b = 0 \textbf{ then } c^* \leftarrow \mathsf{SE}(spk, m^*) \textbf{ else } c^* \stackrel{R}{\leftarrow} \mathcal{C} :\ SA_2^{SO_2^{cca}}(spk, m^*, c^*, s) = b\Big] \ = \ \frac{1}{2} + \epsilon(k).$$

Similarly, we say that Σ is *steganographically secure against replayable adaptive chosen-covertext attacks (SS-RCCA)* with respect to $\mathcal{C}$ if for all probabilistic polynomial-time adversaries (SA_1, SA_2), there exists a negligible function ϵ such that the above equation holds with SO_2^{cca} replaced by SO_2^{rcca}.

Note that this leaves the adversary free to query the decoding oracle with any element of the covertext space *before* the challenge is issued. By definition, an SS-CCA stegosystem is also SS-RCCA.

2.3 Discussion

The relation to public-key cryptosystems. A stegosystem should allow for two parties to communicate over a public channel in such a way that the presence of a message in the conversation cannot be detected by an adversary. It seems natural to conclude from this that the adversary must not learn any useful information about an embedded message, should there be one. The latter property is the subject of cryptography: hiding the content of a message transmitted over a public channel. This motivates our approach of modeling a public-key stegosystem after a public-key cryptosystem in which the ciphertext conforms to a particular covertext distribution.

The most widely accepted formal notion of a public-key cryptosystem secure against an active adversary is *indistinguishability of encryptions against an adaptive chosen-ciphertext attack* (CCA-security) [14] and is equivalent to *non-malleability of ciphertexts* in the same attack model [9, 3]. CCA-security is defined by an experiment with almost the same stages as above, except that the first part of the adversary outputs *two* messages m_0 and m_1, of which one is chosen at random and then encrypted. The resulting value c^*, also called the *target ciphertext*, is returned to the adversary and the adversary has to guess what has been encrypted. In the second query stage, the adversary is allowed to obtain decryptions of *any* ciphertext except for c^*.

This appears to the minimal requirement to make the definition of a cryptosystem meaningful, but it has turned out to be overly restrictive in some cases. For example, consider a CCA-secure cryptosystem where a useless bit is appended to each ciphertext during encryption and that is ignored during decryption. Although this clearly does not affect the security of the cryptosystem, the modified scheme is no longer CCA-secure.

Several authors have relaxed CCA-security to allow for such changes [17, 1, 5]; the weakest one among the relaxed notions is called *replayable CCA-security* or *RCCA-security* [5]. The only difference to CCA-security is that in the second query stage, the adversary is more restricted and does not allow any query that decrypts to either one of the messages m_0 or m_1. The intuition is that such a cryptosystem allows anyone to modify a ciphertext into an equivalent one and therefore “replay” the target ciphertext.

Our notion of an SS-CCA stegosystem is analogous to a CCA-secure cryptosystem, in that it only excludes the target covertext from the queries to the second decoding oracle. Likewise, our notion of an SS-RCCA stegosystem contains a restriction that is reminiscent of an RCCA-secure cryptosystem, by not allowing queries that decode either to the test message or to $\perp$. These similarities are no coincidence: We show in Section 3 that any SS-CCA stegosystem is an CCA-secure public-key cryptosystem, and similarly for their replayable counterparts.

Previous models for steganography. The first published model of a steganographic system is the "Prisoners' Problem" by Simmons [19]. This work addresses the particular situation of message authentication among two communicating parties, where a so-called *subliminal channel* might be used to transport a hidden message in the view of an adversary who tries to detect the presence of a hidden message. Although a subliminal channel in that sense is only made possible by the existence of message authentication in the model, it can be seen as the first formulation of a general model for steganography.

Cachin [4] presented an information-theoretic model for steganography, which was the first to explicitly require that the stegotext distribution is indistinguishable from the covertext distribution to an adversary. Since the model is unconditional, a statistical information measure is used.

Hopper et al. [11] give the first complexity-theoretic model for private-key steganography with passive attacks; they point out that a stegosystem is similar to a cryptosystem whose ciphertext is indistinguishable from a given covertext. In Section 3 we establish such an equivalence formally for public-key systems.

Recently, von Ahn and Hopper [20] have formalized public-key steganography with a passive adversary, i.e., one who can mount a chosen-message attack. The resulting notion is the analogue of a cryptosystem with security against chosen-plaintext attacks (i.e., a cryptosystem with semantic security). They also formalized the notion of a stegosystem that offers security against "attacker-specific" chosen-stegotext attacks; this means that the decoder must know the identity of the encoder, however, and restricts the usefulness of their notion compared to SS-CCA and SS-RCCA.

No satisfying formal model for public-key steganography with active attacks has been published so far, although the subject was discussed by several authors, and some systems with heuristic security have been proposed [8, 2]. A crucial element that seems to make our formalizations useful is the restriction of the stage-two decoding oracle depending on the challenge covertext.

3 Results

This section investigates the relation between SS-(R)CCA stegosystems and (R)CCA-secure public-key cryptosystems. Two results are presented:

1. Any SS-CCA stegosystem is a CCA-secure public-key cryptosystem and, similarly, any SS-RCCA stegosystem is an RCCA-secure public-key cryptosystem.

2. An SS-RCCA stegosystem can be constructed from any RCCA-secure public-key cryptosystem whose ciphertexts are pseudorandom.

We first recall the formal definitions for public-key encryption with CCA- and RCCA-security, respectively. A *public-key cryptosystem* is a triple $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ of probabilistic polynomial-time algorithms. Algorithm K, on input the security parameter k, generates a pair of keys (sk, pk). The encryption and decryption algorithms, E and D, have the property that for any pair (sk, pk) generated by K and for any plaintext message $m \in \{0,1\}^l$, the probability that $\mathsf{D}(1^k, sk, \mathsf{E}(1^k, pk, m)) \neq m$ is negligible in k. (The security parameter is omitted henceforth.)

CCA-security and RCCA-security for a public-key encryption scheme are defined by the following experiment. Consider an adversary defined by two arbitrary polynomial-time algorithms A_1 and A_2. First, a key pair (pk, sk) is generated by K. Next, A_1 is run on input the public key pk and may access a decryption oracle O_1. Oracle O_1 knows the secret key sk, and whenever it receives a ciphertext c, it applies D with key sk to c and returns the result to A_1. When A_1 finishes its execution, it outputs a triple (m_0, m_1, s), where $m_0, m_1 \in \{0,1\}^l$ are two arbitrary messages and s is some additional state information. Now a bit b is chosen at random and m_b is encrypted using E under key pk, resulting in a ciphertext c^*. Algorithm A_2 is given m_0 and m_1, ciphertext c^*, and state s, and has to guess the value of b, i.e., whether m_0 or m_1 has been encrypted. For CCA-security, A_2 may access a decryption oracle O_2^{cca}, which is analogous to O_1 and knows sk, but returns `not-allowed` upon receiving query c^*. For RCCA-security, A_2 may access a decryption oracle O_2^{rcca}, which is identical to O_1^{cca} except that any query that decrypts to one of the messages m_0 and m_1 are answered by `not-allowed`. Finally, A_2 outputs a bit b' as its guess for b.

A secure cryptosystem requires that no efficient adversary can distinguish an encryption of m_0 from an encryption of m_1 except with negligible probability.

Definition 3. [(R)CCA-Security for Public-Key Cryptosystems [3, 5]] Let $\Omega = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be a public-key cryptosystem. We say that Ω is *CCA-secure* if for all polynomial-time adversaries $A = (A_1, A_2)$, there exists a negligible function ϵ such that

$$\Pr\Big[(pk, sk) \leftarrow \mathsf{K};\ (m_0, m_1, s) \leftarrow A_1^{O_1}(pk);\ b \stackrel{R}{\leftarrow} \{0,1\};\\ c^* \leftarrow \mathsf{E}(pk, m_b);\ A_2^{O_2^{cca}}(pk, m_0, m_1, c^*, s) = b\Big] \ = \ \frac{1}{2} + \epsilon(k).$$

We say that Ω is *RCCA-secure* if the same holds with O_2^{cca} replaced O_2^{rcca}.

The following is our first main result.

Theorem 1. *Let $\Sigma = (\mathit{SK}, \mathit{SE}, \mathit{SD})$ denote a public-key stegosystem. If Σ is SS-CCA (SS-RCCA) with respect to some distribution $\mathcal{C}$, then Σ is an CCA-secure (RCCA-secure) public-key cryptosystem.*

Proof. Note first that Σ satisfies the definition of a public-key cryptosystem. We prove that Σ is (R)CCA-secure by a reduction argument. Assume that Σ is not an (R)CCA-secure cryptosystem and hence there exists an (encryption-)adversary (A_1, A_2) that breaks the (R)CCA-security of Σ, i.e., it wins in the experiment of Definition 3 with probability $\frac{1}{2} + \delta(k)$ for some non-negligible function δ. Let $\mathcal{C}$ be an arbitrary distribution. We construct a (stego-)adversary $(\mathit{SA}_1, \mathit{SA}_2)$ against Σ as a stegosystem with respect to $\mathcal{C}$ that has black-box access to (A_1, A_2) as follows.

Key generation: When SA_1 receives a public-key, it invokes A_1 with this key.

First decoding stage: Whenever A_1 queries its decryption oracle O_1 with a ciphertext c, SA_1 passes c on to its decoding oracle SO_1, waits for the response and forwards the response to A_1.

When A_1 halts and outputs (m_0, m_1, s), the stego-adversary SA_1 chooses a random bit b', and outputs $(m_{b'}, (m_0, m_1, b', s))$.

Challenge: A challenge covertext c^* is computed according to the definition of a stegosystem and given to SA_2.

Second decoding stage: SA_2 receives inputs $m_{b'}$, c^*, and (m_0, m_1, b', s) and invokes A_2 on inputs m_0, m_1, c^*, and s. Otherwise, SA_2 behaves in the same way as SA_1 during first decoding stage, forwarding the decryption requests that A_2 makes to O_2 to the respective decoding oracle SO_2^{cca} or SO_2^{rcca}. If the distinction between SO_2^{cca} and SO_2^{rcca} is irrelevant, we simply write SO_2, similarly for the decryption oracle O_2.

Guessing stage: When A_2 outputs a bit b^*, the stego-adversary SA_2 tests if $b^* = b'$ and outputs 0 if true, and 1 otherwise.

We now analyze the environment simulated by the stego-adversary (SA_1, SA_2) to the encryption-adversary (A_1, A_2), and the probability that the stego-adversary can distinguish stegotext from covertext.

Clearly, key generation and the first decoding stage perfectly simulate the decryption oracle to adversary A_1. During the challenge, a random bit b is chosen and a challenge covertext $c^* \leftarrow \mathsf{SE}(pk, m_{b'})$ is computed in case $b = 0$ and $c \overset{R}{\leftarrow} \mathcal{C}$ otherwise.

Note that when $b = 1$, algorithm A_2 and its final output b^* are independent of b'. Hence, we have $\Pr[b' = b^* | b = 1] = \frac{1}{2}$ and the stego-adversary has no advantage over randomly guessing b' in that case.

When $b = 0$, we show that during the second decoding phase, SA_2 emulates the decryption oracle O_2 to A_2 except with negligible probability. We only have to show that A_2 never queries any value that is permitted for decryption oracle O_2 but forbidden for decoding oracle SO_2. Apart from this, the emulation is perfect by definition.

For SS-CCA, a query c' to SO_2^{cca} is not allowed if $c' = c^*$, which means that the query c' is also not allowed for the decryption oracle O_2^{cca} and A_2 will receive the correct answer `not-allowed`, except with negligible probability. For SS-RCCA, a query c' to SO_2^{rcca} is not allowed if $\mathsf{SD}(ssk, c') \in \{m_{b'}, \perp\}$. However, since $\mathsf{D}(sk, c') = m_{b'}$ except with negligible probability by the definition of a public-key cryptosystem, the query c' is also not allowed for the decryption oracle O_2^{rcca} and A_2 will receive the correct answer `not-allowed`, except with negligible probability.

Because A_2 makes at most a polynomial number of queries to the decryption oracle O_2, the probability that at least one of them is allowed for decryption but not allowed for the decoding oracle SO_2 is also negligible. Hence, SA_2 correctly simulates the decryption oracle O_2 to A_2 except with some negligible probability $\epsilon^*(k)$.

Since the encryption-adversary A_2 by assumption breaks the (R)CCA-security of the cryptosystem, and A_2 is independent of b' when $b = 1$ as argued above, we have $\Pr[b' = b^* | b = 0] = \frac{1}{2} + 2\delta(k) - \epsilon^*(k)$. By the definition of SA_2, this is also the probability that the stego-adversary guesses b correctly when $b = 0$. Hence, the overall probability that SA_2 guesses b correctly is $\frac{1}{2} + \delta(k) - \frac{\epsilon^*(k)}{2}$, which exceeds $\frac{1}{2}$ by a non-negligible quantity and shows that Σ is not SS-(R)CCA with respect to any $\mathcal{C}$. □

Theorem 1 shows that an SS-CCA stegosystem is a special case of a CCA-secure public-key cryptosystem, and similarly for their replayable variants. In the converse direction, we show now that some RCCA-secure public-key cryptosystems, namely those with "pseudorandom ciphertexts," can also be used to construct SS-RCCA stegosystems. Constructing an SS-CCA stegosystem from a CCA-secure public-key cryptosystem — or from other assumptions, for that matter — remains an open problem.

In a cryptosystem with pseudorandom ciphertexts, the encryption algorithm outputs a bit string that is indistinguishable from a random string of the same length for any efficient distinguisher that has knowledge of the public key. We make the usual assumption that the encryption of a plaintext of length l always results in a ciphertext of length $\ell(l)$.

Definition 4. [Public-key Cryptosystem with Pseudorandom Ciphertexts [20]] A public-key cryptosystem $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ is said to have *pseudorandom ciphertexts* if for any key pair (sk, pk) generated by K, any $m \in \{0,1\}^l$, and all probabilistic polynomial-time distinguishers A, there exists a negligible function ϵ

such that

$$\Pr\Big[c_0 \leftarrow \mathsf{E}(pk,m);\ c_1 \xleftarrow{R} \{0,1\}^{\ell(l)};\ b \xleftarrow{R} \{0,1\}; A(pk,m,c_b) = b\Big] \ = \ \frac{1}{2} + \epsilon(k).$$

It seems difficult to construct SS-(R)CCA stegosystems for *any* covertext distribution. We show that it is possible for covertexts whose distribution conforms to a sequence of independently repeated experiments. (According to the remark in Section 2.2, this result generalizes to an arbitrary covertext *channel*.) Given a covertext distribution $\mathcal{C}$ and positive t, let $\mathcal{C}^t$ denote the probability distribution consisting of a sequence of t independent repetitions of $\mathcal{C}$.

The next theorem is our second main result. Its proof is the subject of Section 4.

Theorem 2. *SS-RCCA stegosystems with respect to a covertext distribution $\mathcal{C}^t$ for any $\mathcal{C}$ with sufficiently large min-entropy can be efficiently constructed from any RCCA-secure cryptosystem with pseudorandom ciphertexts.*

Theorem 2 leaves us with the task of finding an RCCA-secure cryptosystem with pseudorandom ciphertexts. Such cryptosystems exist under a variety of standard assumptions if one asks for security against a *passive* adversary only, i.e., security against *chosen-plaintext attacks (CPA)*. For example, von Ahn and Hopper [20] demonstrate a scheme that is as secure as RSA and one that is secure under the Decisional Diffie-Hellman (DDH) assumption. It is also straightforward to verify that the generic method of encrypting a single bit by xoring it with the hard-core predicate of a trapdoor one-way permutation has pseudorandom ciphertexts.

But any RCCA-secure cryptosystem can be turned into one with pseudorandom ciphertexts using the following method, suggested by Lindell [12]: Take the ciphertext output by the RCCA-secure encryption algorithm and encrypt it again, using a second cryptosystem with pseudorandom ciphertexts, which is secure against chosen-plaintext attacks. Decryption proceeds analogously, by first applying the decryption operation of the second cryptosystem and then the decryption operation of the RCCA-secure cryptosystem. It can be verified that the composed cryptosystem retains RCCA-security because the stage-two decryption oracle knows both secret keys. This method yields SS-RCCA stegosystems in several models as follows.

By applying the generic construction of a CPA-secure cryptosystem with pseudorandom ciphertexts to a generic non-malleable cryptosystem [9, 16], we conclude that SS-RCCA stegosystems exist under general assumptions.

Corollary 3. *Provided that trapdoor one-way permutations exist, there is an SS-RCCA stegosystem in the common random string model.*

Using the mentioned DDH-based cryptosystem with pseudorandom ciphertexts combined with the Cramer-Shoup cryptosystem [7], we obtain also an efficient SS-RCCA stegosystem in the standard model.

Corollary 4. *Under the Decisional Diffie-Hellman assumption, there is an SS-RCCA stegosystem.*

A more practical cryptosystem with pseudorandom ciphertexts exists also in the random oracle model: the OAEP+ scheme of Shoup [18]. OAEP+ is a CCA-secure cryptosystem based on an arbitrary trapdoor one-way permutation.

Corollary 5. *Provided that trapdoor one-way permutations exist, there is an SS-RCCA stegosystem in the random oracle model.*

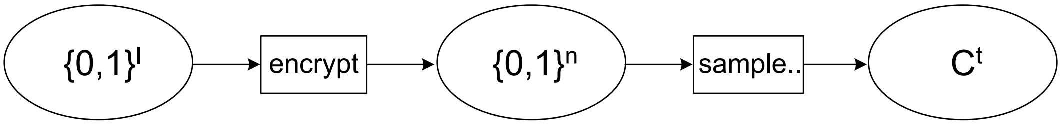

Figure 1: The encoding process of the stegosystem: a message is first encrypted and then embedded using Algorithm sample. The decoding process works analogously in the reverse direction.

4 An SS-RCCA Stegosystem

In this section, we propose a stegosystem that is steganographically secure against replayable adaptive chosen-covertext attacks.

This stegosystem works for any covertext distribution that consists of a sequence of independent repetitions of a base-covertext distribution. Deviating from the notation of Section 2, we denote the base-covertext distribution by $\mathcal{C}$ and the covertext distribution used by the stegosystem by $\mathcal{C}^t = \Pi_{i=1}^t \mathcal{C}$. As noted in Section 2.2, through the introduction of a history, our construction also generalizes to arbitrary covertext channels.

Let $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ be an RCCA-secure public-key cryptosystem with pseudorandom ciphertexts. Suppose its cleartexts are l-bit strings and its ciphertexts are n-bit strings.

A class G of functions $X \to Y$ is called *strongly* 2*-universal* [21] if, for all distinct $x_1, x_2 \in X$ and all (not necessarily distinct) $y_1, y_2 \in Y$, exactly $|G|/|Y|^2$ functions from G take x_1 to y_1 and x_2 to y_2. Such a function family is sometimes simply called a *strongly 2-universal hash function* for brevity.

4.1 Description

The SS-RCCA stegosystem consists of a triple of algorithms $(\mathsf{keygen}, \mathsf{encode}, \mathsf{decode})$. The idea behind it is to encrypt a message using the public-key cryptosystem first and to embed the resulting ciphertext into a covertext sequence, as shown in Figure 1.

The encoding method is based on the following algorithm sample, which samples a base-covertext according to $\mathcal{C}$ such that a given f-bit string b is embedded in it. Under the name "rejection sampler," this algorithm has been suggested previously for steganography [2, 11, 15], but was restricted to embedding single-bit messages only.

Algorithm sample

Input: security parameter k, a function $g : C \to \{0,1\}^f$, and a value $b \in \{0,1\}^f$
Output: a covertext x

$j \leftarrow 0$
repeat
  $x \stackrel{R}{\leftarrow} \mathcal{C}$
  $j \leftarrow j + 1$
until $g(x) = b$ **or** $j = k$
return x

Intuitively, algorithm sample returns a covertext chosen from distribution $\mathcal{C}$, but restricted to that subset of C which is mapped to the given b by g. sample may also fail and return a covertext c with $g(c) \neq b$, but this happens only with negligible probability in k. As will be shown in Section 4.2, when b is a random f-bit string, g is chosen randomly from a 2-universal hash function, and $\mathcal{C}$ has sufficient min-entropy, then the output distribution of sample is statistically close to $\mathcal{C}$.

We now turn to the description of the stegosystem. Let $f \leq \gamma \log k$ for a positive constant $\gamma < 1$ and let $G : C \to \{0,1\}^f$ denote a strongly 2-universal hash function.

Algorithm keygen chooses a random $g \stackrel{R}{\leftarrow} G$ and computes a tuple $(sk, pk) \leftarrow \mathsf{K}$, by running the key generation algorithm of the cryptosystem. The output of keygen is the tuple $(spk, ssk) = ((pk, g), sk)$.

Algorithm encode first encrypts an input message m using the given encryption algorithm E, which outputs in a ciphertext y. Assuming w.l.o.g. that y is an n-bit string such that n is polynomial in k and $n = tf$, encode then repeatedly invokes sample to embed y in pieces of f bits a time into a sequence of t covertext symbols. Formally:

Algorithm encode

Input: security parameter k, a public key *spk* $= (pk, g)$, and a message $m \in \{0,1\}^l$ to encode
Output: a covertext $(c_1, \ldots, c_t)$
1: $y \leftarrow \mathsf{E}(pk, m)$
2: parse y as $y_1 \| y_2 \| \cdots \| y_t$, where $y_i \in \{0,1\}^f$
3: **for** $i = 1$ to t **do**
4: $\quad c_i \leftarrow \mathsf{sample}(k, g, y_i)$
5: return $(c_1, \ldots, c_t)$

Algorithm decode proceeds analogously. From each of the t symbols in the covertext, a string of f bits is extracted by g; then the concatenation of these bit strings is decrypted by D, and the resulting value is returned (this is either an l-bit message or the symbol $\perp$):

Algorithm decode

Input: security parameter k, a secret key *ssk* $= (sk, g)$, and a covertext $(c_1, \ldots, c_t) \in C^t$ to decode
Output: a decoded l-bit message or $\perp$
1: **for** $i = 1$ to t **do**
2: $\quad y_i \leftarrow g(c_i)$
3: $y \leftarrow y_1 \| y_2 \| \cdots \| y_t$
4: $x \leftarrow \mathsf{D}(sk, y)$
5: return x

4.2 Analysis

This section is devoted to an analysis of the above stegosystem. Theorems 6 and 10 below together imply Theorem 2.

Theorem 6. *(keygen, encode, decode) is a valid stegosystem.*

Proof (Sketch). According to Definition 1, the only non-trivial steps are to show that the algorithms are efficient and that

$$\mathsf{decode}(1^k, \mathit{ssk}, \mathsf{encode}(1^k, \mathit{spk}, m)) \;=\; m$$

for all $m \in \{0,1\}^l$ except with negligible probability.

Efficiency follows immediately from the construction, the assumption $f \leq \gamma \log k$, and the efficiency of the public-key cryptosystem.

For reliability, it suffices to analyze the output of encode because the decoding operation is deterministic.

Consider iteration i in Algorithm encode, in which Algorithm sample tries to find a covertext x that is mapped to y_i by g. Because g is chosen from a strongly 2-universal class of hash functions, the probability that in any particular iteration of sample, an x is chosen with $g(x) \neq y_i$, is $1 - 2^{-f}$.

Thus, since the k iterations in sample are independent, sample returns c with $g(c) \neq y_i$ only with some negligible probability $\epsilon(k)$ provided that $f \leq \gamma \log k$.

Hence, by the union bound, the probability that any iteration of Algorithm encode fails to embed the correct value is at most $t\epsilon(k)$, which is negligible. □

Before we can analyze the security of the stegosystem (keygen, encode, decode), we investigate the output distribution of Algorithm sample and derive the following result that may be of independent interest. It shows that the distribution of the output from Algorithm sample is statistically close to $\mathcal{C}$ when sample is run with uniformly chosen inputs. The result also generalizes a theorem of Reyzin and Russell [15].

Let sample be run with independently chosen $b \overset{R}{\leftarrow} \{0,1\}^f$ and $g \overset{R}{\leftarrow} G$, and denote by $\mathcal{S}(k)$ the distribution of its output.

Proposition 7. *If the min-entropy of the covertext distribution $\mathcal{C}$ is large enough compared to f, then the statistical distance between $\mathcal{S}(k)$ and $\mathcal{C}$ is negligible; in particular, there exists a positive constant $\lambda < 1$ such that for all sufficiently large k*

$$\|\mathcal{S}(k) - \mathcal{C}\| \;<\; 2^{f - H\infty(\mathcal{C})} + \lambda^k.$$

The proof of this result is based on Lemmas 8 and 9 below. Given a function g used by Algorithm sample and a value b, define

$$\gamma(g,b) = \Pr[x \overset{R}{\leftarrow} \mathcal{C} : g(x) = b].$$

Let $\epsilon(g,b) = 1 - \gamma(g,b)$.

Lemma 8. *For a given function g and a value b, the probability that Algorithm* sample *outputs a particular c is*

$$\Pr[\mathsf{sample}(\mathcal{C}, g, b, k) = c] \;=\; \begin{cases} \big(1 - \epsilon(g,b)^k\big)\frac{\Pr_{\mathcal{C}}[c]}{\gamma(g,b)} & \textit{if } g(c) = b \\ \epsilon(g,b)^k \frac{\Pr_{\mathcal{C}}[c]}{\epsilon(g,b)} & \textit{otherwise} \end{cases}$$

Proof. The probability of a value c under distribution $\mathcal{C}$ conditioned on the event $g(\mathcal{C}) = b$ is equal to $\Pr_{\mathcal{C}}[c]/\gamma(g,b)$ if $g(c) = b$ and 0 otherwise; similarly, the probability of c under the conditional distribution of $\mathcal{C}$ given $g(\mathcal{C}) \neq b$ is $\Pr_{\mathcal{C}}[c]/\epsilon(g,b)$ if $g(c) \neq b$ and 0 otherwise. By construction, the second case, i.e., sample outputs c with $g(c) \neq b$, occurs if and only if the loop terminated with $j = k$; this happens with probability $\epsilon(g,b)^k$ because the realizations of $\mathcal{C}$ are independent. The first case covers any other outcome of the algorithm. □

Lemma 9. *For every distribution $\mathcal{C}$, there exists $0 < \lambda < 1$ such that for all sufficiently large k and all $c \in C$,*

$$2^{-f}\big(1 - \lambda^k\big)\,\frac{\Pr_{\mathcal{C}}[c]}{|G|}\sum_{g \in G}\frac{1}{\gamma(g, g(c))} \;<\; \Pr_{\mathcal{S}(k)}[c] \;<\; 2^{-f}\big(1 + \lambda^k\big)\,\frac{\Pr_{\mathcal{C}}[c]}{|G|}\sum_{g \in G}\frac{1}{\gamma(g, g(c))}. \quad (1)$$

Proof.

$$
\begin{aligned}
\Pr_{\mathcal{S}(k)}[c] &= \Pr[b \stackrel{R}{\leftarrow} B; g \stackrel{R}{\leftarrow} G; x \stackrel{R}{\leftarrow} \textsf{sample}(\mathcal{C}, b, g, k) : x = c] \\
&= 2^{-f} \sum_{b \in B} \frac{1}{|G|} \sum_{g \in G} \Pr[x \stackrel{R}{\leftarrow} \textsf{sample}(\mathcal{C}, b, g, k) : x = c] \\
&= 2^{-f} \frac{1}{|G|} \sum_{b \in B} \Bigg(\sum_{g : g(c) = b} \big(1 - \epsilon(g,b)^k\big) \frac{\Pr_{\mathcal{C}}[c]}{\gamma(g,b)} + \sum_{g : g(c) \neq b} \epsilon(g,b)^k \frac{\Pr_{\mathcal{C}}[c]}{\epsilon(g,b)} \Bigg) \qquad (2) \\
&= 2^{-f} \frac{\Pr_{\mathcal{C}}[c]}{|G|} \sum_{g \in G} \Bigg(\sum_{b : b = g(c)} \frac{1 - \epsilon(g,b)^k}{\gamma(g,b)} + \sum_{b : b \neq g(c)} \epsilon(g,b)^{k-1} \Bigg) \qquad (3) \\
&= 2^{-f} \frac{\Pr_{\mathcal{C}}[c]}{|G|} \sum_{g \in G} \Bigg(\frac{1 - \epsilon(g, g(c))^k}{\gamma(g, g(c))} + \sum_{b : b \neq g(c)} \epsilon(g,b)^{k-1} \Bigg) \qquad (4)
\end{aligned}
$$

where (2) follows from Lemma 8, (3) from switching the order of summation, and (4) from noting that the first sum contains only the term $b = g(c)$.

Recall that $\Pr_{\mathcal{C}}[c] > 0$ for all $c \in C$ and that $2^f < k$. Hence, $0 < \epsilon(g,b) < 1$ and there exists $0 < \lambda < 1$ such that for all sufficiently large k,

$$
\Bigg| \epsilon(g, g(c))^k + \gamma(g, g(c)) \sum_{b : b \neq g(c)} \epsilon(g, g(c))^{k-1} \Bigg| < \lambda^k .
$$

The lemma follows from combining this with (4). □

Proof of Proposition 7. For a particular function g and a covertext c, define $A_c(g) = \gamma(g, g(c))$ and consider $A_c(g)$ as a random variable induced by the random choice with uniform distribution of g from G. The expectation of $A_c(g)$ is

$$
\begin{aligned}
\mathrm{E}[A_c(g)] &= \sum_{g \in G} \Pr_G[g] \gamma(g, g(c)) \\
&= \Pr[g \stackrel{R}{\leftarrow} G; x \stackrel{R}{\leftarrow} \mathcal{C} : g(x) = g(c)] \\
&= \Pr[x \stackrel{R}{\leftarrow} \mathcal{C} : x = c] + \Pr\big[g \stackrel{R}{\leftarrow} G; x \stackrel{R}{\leftarrow} \mathcal{C}|_{C \setminus \{c\}} : g(x) = g(c)\big] \big(1 - \Pr[x \stackrel{R}{\leftarrow} \mathcal{C} : x = c]\big) \\
&\leq p_{\max}(\mathcal{C}) + 2^{-f} = 2^{-H_\infty(\mathcal{C})} + 2^{-f}, \qquad (5)
\end{aligned}
$$

where $\mathcal{C}|_{C \setminus \{c\}}$ denotes the conditional distribution of $\mathcal{C}$ restricted to $C \setminus \{c\}$ and the inequality follows from the definition of $p_{\max}$ and from the 2-universality of G.

Note that the bound of Lemma 9 involves the expected value of $(A_c(g))^{-1}$ (over the random choice of g). The Jensen inequality [6] states that for any convex function f applied to a random variable X, the expected value of $f(X)$ is at least as big as f applied to the expected value of X. Thus, $\mathrm{E}\big[(A_c(g))^{-1}\big] \geq$

$\big(\mathrm{E}[A_c(g)]\big)^{-1}$ for all $c \in C$. We get

$$\begin{aligned}
\|\mathcal{C} - \mathcal{S}(k)\| &= \sum_{c:\Pr_{\mathcal{C}}[c] > \Pr_{\mathcal{S}(k)}[c]} \Pr_{\mathcal{C}}[c] - \Pr_{\mathcal{S}(k)}[c] \\
&< \sum_{c:\Pr_{\mathcal{C}}[c] > \Pr_{\mathcal{S}(k)}[c]} \Big(\Pr_{\mathcal{C}}[c] \Big(1 - \frac{1-\lambda^k}{2^f |G|} \sum_{g \in G} \frac{1}{\gamma(g, g(c))} \Big) \Big) && (6) \\
&\le \sum_{c:\Pr_{\mathcal{C}}[c] > \Pr_{\mathcal{S}(k)}[c]} \Big(\Pr_{\mathcal{C}}[c] \Big(1 - \frac{1-\lambda^k}{2^f} \mathrm{E}\big[(A_c(g))^{-1} \big] \Big) \Big) && (7) \\
&\le \sum_{c:\Pr_{\mathcal{C}}[c] > \Pr_{\mathcal{S}(k)}[c]} \Big(\Pr_{\mathcal{C}}[c] \Big(1 - \frac{1-\lambda^k}{2^f (2^{-f} + 2^{-H_\infty(\mathcal{C})})} \Big) \Big) && (8) \\
&\le 1 - \frac{1-\lambda^k}{1 + 2^{f - H_\infty(\mathcal{C})}} \\
&\le 2^{f - H\infty(\mathcal{C})} + \lambda^k,
\end{aligned}$$

where (6) follows from Lemma 9, (7) from the Jensen inequality and from the definition of $A_c(g)$, and (8) from (5). □

Theorem 10. *For a covertext distribution $\mathcal{C}^t$ such that $\mathcal{C}$ has sufficiently large min-entropy and provided that $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ is an RCCA-secure public-key cryptosystem with pseudorandom ciphertexts, the stegosystem $(\mathsf{keygen}, \mathsf{encode}, \mathsf{decode})$ is SS-RCCA.*

Proof (Sketch). We prove that the stegosystem $(\mathsf{keygen}, \mathsf{encode}, \mathsf{decode})$ is SS-RCCA by a reduction argument. Assume that it is not SS-RCCA and and hence there exists a (stego-)adversary (SA_1, SA_2) that succeeds in the experiment of Definition 2 with probability $\frac{1}{2} + \delta(k)$ for some non-negligible function δ. We construct an (encryption-)adversary (A_1, A_2) that has black-box access to (SA_1, SA_2) and breaks the RCCA-security of $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ as follows.

Key generation: When A_1 receives a public-key *pk* generated by K, it chooses $g \stackrel{R}{\leftarrow} G$, computes $spk \leftarrow (pk, g)$, and invokes SA_1 with *spk*.

First decryption stage: When SA_1 sends a query $(c_1, \ldots, c_t)$ to its decoding oracle SO_1, then A_1 computes $y \leftarrow y_1 \| y_2 \| \cdots \| y_t$ for $y_i \leftarrow g(c_i)$, gives y to its decryption oracle O_1, waits for the response and forwards the response to SA_1.

Challenge: When SA_1 halts and outputs (m^*, s), the encryption-adversary A_1 chooses an arbitrary plaintext message $m' \in \{0,1\}^l$ and outputs (m^*, m', g). According to the definition of a public-key cryptosystem, a challenge ciphertext y^* is computed. Now A_2 is invoked with inputs *pk*, m^*, m', y^*, and g. It parses y^* as a sequence $y_1^* \| y_2^* \| \cdots \| y_t^*$ of f-bit strings, computes $c_i^* \leftarrow \mathsf{sample}(k, g, y_i^*)$ for $i = 1, \ldots, t$, and invokes SA_2 with inputs (pk, g), m^*, $(c_1^*, \ldots, c_t^*)$, and s.

Second decryption stage: A_2 behaves in the same way as A_1 during first decryption stage: It computes a ciphertext y from any decoding request that SA_2 makes as above, submits y to the decryption oracle O_2, and returns the answer to SA_2.

Guessing stage: When SA_2 outputs a bit b^*, indicating its guess as to whether message m^* is contained in the challenge covertext $(c_1^*, \ldots, c_t^*)$, the encryption-adversary A_2 returns b^* as its own guess of whether m^* or m' is encrypted in y^*.

We now analyze the environment simulated by the encryption-adversary (A_1, A_2) to the stego-adversary (SA_1, SA_2) and the probability that the encryption-adversary can distinguish the encrypted messages.

Clearly, during key generation and the first decoding stage, the simulation for the stego-adversary SA_1 is perfect. During the encoding stage, a random bit b is chosen according to Definition 3 and the challenge ciphertext is computed as $y^* \leftarrow \mathsf{E}(pk, m^*)$ if $b = 0$ and $y^* \leftarrow \mathsf{E}(pk, m')$ if $b = 1$.

When $b = 0$, then, according to the definition of A_1, the challenge covertext c^* is computed in the same way as expected by the stego-adversary in the experiment of Definition 2 and the simulation is perfect.

When $b = 1$, however, SA_2 expects $(c_1^*, \ldots, c_t^*)$ to be a random covertext drawn according to $\mathcal{C}^t$, but receives $c_i^* = \mathsf{sample}(k, g, y_i^*)$ for $i = 1, \ldots, t$ instead, where the concatenation of the y_i^* is an encryption of m' under key pk with E.

Proposition 7 implies that for every $i \in \{1, \ldots, t\}$, the statistical distance between $\mathcal{C}$ and the distribution of c_i^* as computed by Algorithm sample when run with input a *uniformly chosen* f-bit string is bounded by a negligible quantity $\epsilon_1^*(k)$. Furthermore, since the cryptosystem $(\mathsf{K}, \mathsf{E}, \mathsf{D})$ has pseudorandom ciphertexts, for every distinguisher SA_2 there exists a negligible quantity $\epsilon_2^*(k)$ such that its advantage (over guessing randomly) in distinguishing between y_i^* as used by A_2 and the uniform distribution on f-bit strings is at most $\epsilon_2^*(k)$.

By combining these two facts, it follows that the behavior of the stego-adversary SA_2 who observes $(c_1^*, \ldots, c_t^*)$ in the simulation when $b = 1$ does not differ from its behavior in experiment of Definition 2, where it observes covertext $\mathcal{C}^t$, with more than probability $\epsilon^*(k) = t\big(\epsilon_1^*(k) + \epsilon_2^*(k)\big)$.

By definition, the output of the encryption-adversary A_2 is the same as that of the stego-adversary SA_2. Since SA_2 succeeds with probability $\frac{1}{2} + \delta(k)$ in attacking the stegosystem and since the simulated view of SA_2 is correct except with probability $\epsilon^*(k)$ when $b = 1$, the probability that SA_2 breaks RCCA-security is $\frac{1}{2} + \delta(k) - \frac{\epsilon^*(k)}{2}$, which exceeds $\frac{1}{2}$ by a non-negligible quantity and establishes the theorem. □

References

[1] J. H. An, Y. Dodis, and T. Rabin, "On the security of joint signatures and encryption," in *Advances in Cryptology: EUROCRYPT 2002* (L. Knudsen, ed.), vol. 2332 of *Lecture Notes in Computer Science*, Springer, 2002.

[2] R. J. Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, May 1998.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Advances in Cryptology: CRYPTO '98* (H. Krawczyk, ed.), vol. 1462 of *Lecture Notes in Computer Science*, Springer, 1998.

[4] C. Cachin, "An information-theoretic model for steganography," in *Information Hiding, 2nd International Workshop* (D. Aucsmith, ed.), vol. 1525 of *Lecture Notes in Computer Science*, pp. 306–318, Springer, 1998.

[5] R. Canetti, H. Krawczyk, and J. Nielsen, "Relaxing chosen-ciphertext security," in *Advances in Cryptology: CRYPTO 2003* (D. Boneh, ed.), vol. 2729 of *Lecture Notes in Computer Science*, Springer, 2003.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.

[7] R. Cramer and V. Shoup, “A practical public-key cryptosystem provably secure against adaptive chosen-ciphertext attack,” in *Advances in Cryptology: CRYPTO ’98* (H. Krawczyk, ed.), vol. 1462 of *Lecture Notes in Computer Science*, Springer, 1998.

[8] S. Craver, “On public-key steganography in the presence of an active warden,” in *Information Hiding, 2nd International Workshop* (D. Aucsmith, ed.), vol. 1525 of *Lecture Notes in Computer Science*, pp. 355–368, Springer, 1998.

[9] D. Dolev, C. Dwork, and M. Naor, “Non-malleable cryptography,” *SIAM Journal on Computing*, vol. 30, no. 2, pp. 391–437, 2000.

[10] S. Goldwasser and S. Micali, “Probabilistic encryption,” *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, 1984.

[11] N. J. Hopper, J. Langford, and L. von Ahn, “Provably secure steganography,” in *Advances in Cryptology: CRYPTO 2002* (M. Yung, ed.), vol. 2442 of *Lecture Notes in Computer Science*, Springer, 2002.

[12] Y. Lindell. Personal communication, Jan. 2004.

[13] T. Mittelholzer, “An information-theoretic approach to steganography and watermarking,” in *Information Hiding, 3rd International Workshop, IH’99* (A. Pfitzmann, ed.), vol. 1768 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, 1999.

[14] C. Rackoff and D. R. Simon, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” in *Advances in Cryptology: CRYPTO ’91* (J. Feigenbaum, ed.), vol. 576 of *Lecture Notes in Computer Science*, pp. 433–444, Springer, 1992.

[15] L. Reyzin and S. Russell, “More efficient provably secure steganography.” Cryptology ePrint Archive, Report 2003/093, 2003. `http://eprint.iacr.org/`.

[16] A. Sahai, “Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security,” in *Proc. 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 543–553, 1999.

[17] V. Shoup, “A proposal for an iso standard for public key encryption.” Cryptology ePrint Archive, Report 2001/112, 2001. `http://eprint.iacr.org/`.

[18] V. Shoup, “OAEP reconsidered,” *Journal of Cryptology*, vol. 15, no. 4, pp. 223–249, 2002.

[19] G. J. Simmons, “The prisoners’ problem and the subliminal channel,” in *Advances in Cryptology: Proceedings of Crypto 83* (D. Chaum, ed.), pp. 51–67, Plenum Press, 1984.

[20] L. von Ahn and N. J. Hopper, “Public-key steganography.” To appear in Proc. Eurocrypt 2004; preliminary version in Cryptology ePrint Archive, Report 2003/233, Nov. 2004.

[21] M. N. Wegman and J. L. Carter, “New hash functions and their use in authentication and set equality,” *Journal of Computer and System Sciences*, vol. 22, pp. 265–279, 1981.

[22] J. Zöllner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke, and G. Wolf, “Modeling the security of steganographic systems,” in *Information Hiding, 2nd International Workshop* (D. Aucsmith, ed.), vol. 1525 of *Lecture Notes in Computer Science*, pp. 344–354, Springer, 1998.