

Digital Steganography*

Christian Cachin

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

February 17, 2005

1 Introduction

Steganography is the art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography means “covered writing” in Greek. As the goal of steganography is to hide the *presence* of a message and to create a covert channel, it can be seen as the complement of cryptography, whose goal is to hide the *content* of a message.

A famous illustration of steganography is Simmons’ “Prisoners’ Problem” [10]: Alice and Bob are in jail, locked up in separate cells far apart from each other, and wish to devise an escape plan. They are allowed to communicate by means of sending messages via trusted couriers, provided they do not deal with escape plans. But the couriers are agents of the warden Eve (who plays the role of the adversary here) and will leak all communication to her. If Eve detects any sign of conspiracy, she will thwart the escape plans by transferring both prisoners to high-security cells from which nobody has ever escaped. Alice and Bob are well aware of these facts, so that before getting locked up, they have shared a secret codeword that they are now going to exploit for embedding a hidden information into their seemingly innocent messages. Alice and Bob succeed if they can exchange information allowing them to coordinate their escape and Eve does not become suspicious.

According to the standard terminology of information hiding [8], a legitimate communication among the prisoners is called *coverttext*, and a message with embedded hidden information is called *stegotext*. The distributions of *coverttext* and *stegotext* are known to the warden Eve because she knows what constitutes a legitimate communication among prisoners and which tricks they apply to add a hidden meaning to innocently looking messages.

The algorithms for creating *stegotext* with an embedded message by Alice and for decoding the message by Bob are collectively called a *stegosystem*. A stegosystem should hide the embedded message at least as well as an encryption scheme since it may be enough for the adversary to learn only a small amount of information about the embedded message to conclude that Alice and Bob are conspiring. But steganography requires more than that. The ciphertext generated by most encryption schemes resembles a sequence of random bits, and this is very likely to raise the suspicion of Eve. Instead, *stegotext* should “look” just like innocent *coverttext* even though it contains a hidden message.

*A survey prepared for the *Encyclopedia of Cryptography and Security*

This intuition forms the basis of the recently developed formal approach to steganography [3, 6, 5, 2, 11]. It views a stegosystem as a cryptosystem with the additional property that its output, i.e., the stegotext, is not distinguishable from coartext to the adversary.

Formally, a stegosystem consists of a triple of algorithms for key generation, message encoding, and message decoding, respectively. In the symmetric-key setting considered here, the output of the key generation algorithm is given only to Alice and to Bob.

The coartext is modeled by a distribution \mathcal{C} over a given set C . The coartext may be given explicitly as a list of values or implicitly as an oracle that returns a sample of \mathcal{C} upon request. A stegosystem that does not require explicit knowledge of the coartext distribution is called *universal*.

A more general model of a coartext *channel* has also been proposed in the literature [5], which allows to model dependencies among repeated uses of the same coartext source. A channel consists of 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. The assumption is that the channel oracle can be queried with an arbitrary prefix of a possible channel output, i.e., its past “history,” and it will return the next symbol according to the channel distribution. In order to simplify the presentation, channels are not considered further here, but all definitions and constructions mentioned below can be readily extended to coartext channels.

We borrow the complexity-theoretic notions of *probabilistic polynomial-time algorithms* and *negligible functions*, in terms of a security parameter n , from modern cryptography [4].

Definition 1 (Stegosystem). Let \mathcal{C} be a distribution on a set C of *coartexts*. A *stegosystem* is a triple of probabilistic polynomial-time algorithms $(\mathbf{SK}, \mathbf{SE}, \mathbf{SD})$ with the following properties.

- The *key generation algorithm* \mathbf{SK} takes as input the security parameter n and outputs a bit string sk , called the *[stego] key*.
- The *steganographic encoding algorithm* \mathbf{SE} takes as inputs the security parameter n , the stego key sk and a *message* $m \in \{0, 1\}^l$ to be embedded and outputs an element c of the coartext space C , which is called *stegotext*. The algorithm may access the coartext distribution \mathcal{C} .
- The *steganographic decoding algorithm* \mathbf{SD} takes as inputs the security parameter n , the stego key sk , and an element c of the coartext space 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 \mathbf{SD} has determined that no message is embedded in c .

For all sk output by $\mathbf{SK}(1^n)$ and for all $m \in \{0, 1\}^l$, the probability that $\mathbf{SD}(1^n, sk, \mathbf{SE}(1^n, sk, m)) \neq m$ must be negligible in n .

Note that the syntax of a stegosystem as defined above is equivalent to that of a (symmetric-key) cryptosystem, except for the presence of the coartext distribution. The probability that the decoding algorithm outputs the correct embedded message is called the *reliability* of a stegosystem.

2 Defining Security

The security of a stegosystem is defined in terms of an experiment that measures the capability of the adversary to detect the presence of an embedded message. In a secure stegosystem, Eve cannot distinguish whether Alice is sending legitimate coartext or stegotext.

The attack considered here is a *chosen-message attack*, where the adversary may influence the embedded message but has otherwise no access to the encoding and decoding functions. It parallels the notion of a chosen-plaintext attack against a cryptosystem.

Consider an adversary defined by a pair of algorithms (SA_1, SA_2) . The experiment consists of four stages.

1. A key sk is generated by running the key generation algorithm **SK**.
2. Algorithm SA_1 is run with input the security parameter n ; 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. SA_1 has access to the covert distribution \mathcal{C} .
3. A bit b is chosen at random and a *challenge covert* c^* is determined depending on it: If $b = 0$ then $c^* \leftarrow \mathbf{SE}(sk, m^*)$ (c^* becomes a steganographic encoding of m^*) otherwise $c^* \xleftarrow{R} \mathcal{C}$ (c^* is chosen randomly according to \mathcal{C}).
4. Algorithm SA_2 is run with inputs n, c^*, m^* , and s , and outputs a bit b' . The goal of SA_2 is to guess the value of b , i.e., to determine whether the message m^* has been embedded in c or whether c has simply been chosen according to \mathcal{C} .

The adversary succeeds to distinguish stegotext from covert if $b' = b$ in the above experiment. Since it is trivial to achieve $\Pr[b' = b] = \frac{1}{2}$, what actually counts is the adversary's advantage above randomly guessing b . Formally, we define the *advantage* of adversary (SA_1, SA_2) to be

$$\Pr \left[sk \leftarrow \mathbf{SK}; (m^*, s) \leftarrow SA_1(1^n); b \xleftarrow{R} \{0, 1\}; \right. \\ \left. \mathbf{if } b = 0 \mathbf{ then } c^* \leftarrow \mathbf{SE}(1^n, sk, m^*) \mathbf{ else } c^* \xleftarrow{R} \mathcal{C} : SA_2(1^n, c^*, m^*, s) = b \right] - \frac{1}{2}.$$

Depending on the detection capabilities ascribed to the adversary in terms of its computational power, and depending on its advantage in distinguishing stegotext from covert, one obtains the following security notions for stegosystems:

Perfectly secure steganography: The adversary is an arbitrary, unbounded algorithm and has advantage equal to 0.

Statistically secure steganography: The adversary is an arbitrary, unbounded algorithm and has only negligible advantage (in n).

Computationally secure steganography: The adversary is an arbitrary probabilistic, polynomial-time computable algorithm and has only negligible advantage in n .

The first two notions are collectively known as *information-theoretically secure steganography* and are discussed in Section 3 (cf., information-theoretic security). The third notion, computational security, is discussed in Section 4 (cf., computational security).

This discussion has so far assumed that the sender and the receiver share the same secret key before communicating (cf., symmetric cryptosystem). Departing from that assumption, it is also possible to define *public-key steganography* with computational security. This notion is analogous to public-key cryptography and is the subject of Section 5 (cf., asymmetric cryptosystem).

For the scope of this survey, the adversary is limited to passive attacks. In the Prisoners Problem, this means that the couriers may not change the messages communicated between Alice and Bob and that Eve may not send a message generated by herself to Bob and observe his reaction to it. However, the adversary may influence the messages to be embedded; for example, the warden Eve may determine the details of Alice and Bob's escape plan by choosing to confine them in particular cells.

This survey is about the formal approach to steganography and about stegosystems that offer provable security. An overview of steganography with heuristic security and of the history of steganography is given by Anderson and Petitcolas [1].

What distinguishes steganography from other forms of information hiding is the focus on merely detecting the *presence* of a hidden message. *Watermarking* and *fingerprinting* are two different problems of information hiding, where the existence of a hidden message is public knowledge. The focus in these areas is on hiding the message in perceptual data from an observer that is typically a human, and on embedding the message robustly so that it cannot be removed without significantly distorting the data itself. The difference between watermarking and fingerprinting is that watermarking supplies digital objects with an identification of origin and all objects are marked in the same way; fingerprinting, conversely, attempts to identify individual copies of an object by means of embedding a unique marker in every copy that is distributed to a user.

3 Information-theoretically Secure Steganography

Definition 2 (Perfect Security). Given a covertext distribution \mathcal{C} , a stegosystem (SK, SE, SD) is called *perfectly secure* with respect to \mathcal{C} if for any adversary (SA_1, SA_2) with unbounded computational power, the advantage in the experiment above is equal to 0.

Perfect security for a stegosystem parallels Shannon’s notion of *perfect security* for a cryptosystem [9] (cf., Shannon’s model). The requirement that every adversary has no advantage implies that the distributions of the challenge c^* are equal in the two cases where it was generated from SE (when $b = 0$) and sampled from \mathcal{C} (when $b = 1$). Hence, the adversary obtains *no* information about b because she only observes the challenge c^* and the distribution of c^* is statistically independent of b . Perfectly secure stegosystems were defined by Cachin [3].

Perfectly secure stegosystems exist only for a very limited class of covertext distributions. For example, if the covertext distribution is uniform, the one-time pad is a perfectly secure stegosystem as follows.

Assume the covertext \mathcal{C} is uniformly distributed over the set of n -bit strings for some positive n and let Alice and Bob share an n -bit key sk with uniform distribution. The encoding function computes the bitwise XOR of the n -bit message m and sk , i.e., $SE(1^n, sk, m) = m \oplus sk$; Bob can decode this by computing $SD(1^n, sk, c) = c \oplus sk$. The resulting stegotext is uniformly distributed in the set of n -bit strings. The one-time pad stegosystem is used like this in visual cryptography [7].

For covertext distributions that do not admit perfectly secure stegosystems, one may still achieve the following security notion.

Definition 3 (Statistical Security). Given a covertext distribution \mathcal{C} , a stegosystem (SK, SE, SD) is called *statistically secure* with respect to \mathcal{C} if for all adversaries (SA_1, SA_2) with unbounded computational power, there exists a negligible function ϵ such that the advantage in the experiment above is at most $\epsilon(n)$.

Statistical security for stegosystems may equivalently be defined by requiring that for any sk and any m , the statistical distance between the probability distribution generated by $SE(1^n, sk, m)$ and the covertext distribution is negligible.

Definition 3 was first proposed by Katzenbeisser and Petitcolas [6]. A very similar notion was defined by Cachin [3], using relative entropy between the stegotext and covertext distributions for quantifying the difference between them.

Here is a simple example of a statistically secure stegosystem, adopted from [3]. It is representative for a class of practical stegosystems that embed information in a digital image by modifying the least

significant bit of every pixel representation [1]. Suppose that the cover space C is the set of n -bit strings with (C_0, C_1) being a partition of C and with distribution \mathcal{C} such $|\Pr[c \xleftarrow{R} \mathcal{C} : c \in C_0] - \Pr[c \xleftarrow{R} \mathcal{C} : c \in C_1]| = \delta(n)$ for some negligible δ . Then there is a stegosystem for a one-bit message m using a one-bit secret key sk . The encoding algorithm **SE** computes $s \leftarrow m \oplus sk$ and outputs $c \xleftarrow{R} C_s$. Decoding works without error because $m = 0$ if and only if $c \in C_{sk}$. It is easy to see that the encoding provides perfect secrecy for m and that the stegosystem is statistically secure. Note, however, that finding the partition for a given distribution is an NP-hard combinatorial optimization problem.

There exist also statistically secure *universal* stegosystems, where the covertext distribution is only available as a sampling oracle. Information-theoretically secure stegosystems suffer from the same drawback as cryptosystems with unconditional security in the sense that the secret key may only be used once. This is not the case for computational security considered next.

4 Computationally Secure Steganography

Definition 4 (Computational Security). Given a covertext distribution \mathcal{C} , a stegosystem (SK, SE, SD) is called *computationally secure* 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 advantage in the experiment above is at most $\epsilon(n)$.

The notion was formalized independently by Katzenbeisser and Petitcolas [6] and by Hopper, Langford, and von Ahn [5]. The latter work also presented the following construction of a computationally secure, universal stegosystem. It illustrates a popular encoding method that does not rely on knowledge of the covertext distribution, which is also used by some practical stegosystems.

The encoding method is based on an algorithm **sample**, which samples a covertext according to \mathcal{C} such that a given bit string b of length $f = O(\log |C|)$ is embedded in it.

Algorithm sample

Input: security parameter n , a function $g : C \rightarrow \{0, 1\}^f$, and a value $b \in \{0, 1\}^f$

Output: a covertext x

- 1: $j \leftarrow 0$
 - 2: **repeat**
 - 3: $x \xleftarrow{R} \mathcal{C}$
 - 4: $j \leftarrow j + 1$
 - 5: **until** $g(x) = b$ **or** $j = n$
 - 6: **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 n .

Suppose $\{G_k\}$ is a pseudorandom function family indexed by k , with domain $\{0, 1\} \times C$ and range $\{0, 1\}^f$. (It can be thought of as a pair (G_0, G_1) of independent pseudorandom functions.) The secret key of the stegosystem consists of a randomly chosen k . The encoding algorithm $\mathbf{SE}(1^n, k, m)$ for an f -bit message m first “encrypts” m to $y \leftarrow G_k(0, c_0) \oplus m$ for a public constant $c_0 \in C$. Note that y is the ciphertext of a symmetric-key encryption of m and is computationally indistinguishable from a random f -bit string. This value y is then embedded by computing a stegotext $c \leftarrow \mathbf{sample}(n, G_k(1, \cdot), y)$. It can be shown that when \mathcal{C} is sufficiently random, as measured in terms of min-entropy, the output distribution of **sample** is statistically close to \mathcal{C} [5, 2].

The decoding algorithm $\text{SD}(1^n, k, c)$ outputs $m' \leftarrow G_k(1, c) \oplus G_k(0, c_0)$; it is easy to show that m' is equal to the message that was embedded using SE except with negligible probability.

This stegosystem is an extension of the example given above for statistical security. In fact, when G is a universal hash function and the encryption is realized using a one-time pad, this is a universal stegosystem with statistical security.

5 Public-key Steganography

What if Alice and Bob did not have the time to agree on a secret key before being imprisoned? They cannot use any of the stegosystems presented so far because that would require them to share a common secret key. Fortunately, steganography is also possible without shared secrets, only with public keys, similar to public-key cryptography. The only requirement is that Bob's public key becomes known to Alice in a way that is not detectable by Eve.

Formally, a public-key stegosystem consists of a triple of algorithms for key generation, message encoding, and message decoding like a (secret-key) stegosystem, but the key generation algorithm now outputs a stego key pair (spk, ssk) . The public key spk is made available to the adversary and is the only key needed by the encoding algorithm SE . The decoding algorithm SD needs the secret key ssk as an additional input.

Definition 5 (Public-key Stegosystem). Let \mathcal{C} be a distribution on a set C of *coverttexts*. A *public-key stegosystem* is a triple of probabilistic polynomial-time algorithms $(\text{SK}, \text{SE}, \text{SD})$ with the following properties.

- The *key generation algorithm* SK takes as input the security parameter n and outputs a pair of bit strings (spk, ssk) , called the *[stego] public key* and the *[stego] secret key*.
- The *steganographic encoding algorithm* SE takes as inputs the security parameter n , the stego public key spk and a *message* $m \in \{0, 1\}^l$ and outputs a coverttext $c \in C$.
- The *steganographic decoding algorithm* SD takes as inputs the security parameter n , the stego secret key ssk , and a coverttext $c \in C$, and outputs either a message $m \in \{0, 1\}^l$ or a special symbol \perp .

For all (spk, ssk) output by the key generation algorithm and for all $m \in \{0, 1\}^l$, the probability that $\text{SD}(1^n, ssk, \text{SE}(1^n, spk, m)) \neq m$ must be negligible in n .

Security is defined analogously to the experiment of Section 2, with the difference that the public key spk is additionally given to the adversary algorithms SA_1 and SA_2 and that the challenge coverttext is computed using spk only. With these modifications, a public-key stegosystem $(\text{SK}, \text{SE}, \text{SD})$ is called *secure against chosen-plaintext attacks* if it is computationally secure according to Definition 4.

Secure public-key stegosystems can be constructed using the method of Section 4, but with the pseudorandom function G_0 (which is used for “encryption”) replaced by a public-key cryptosystem that has almost uniform ciphertexts. This property means that the output of the encryption algorithm is computationally indistinguishable from a uniform bit string of the same length.

The definition and several constructions of public-key stegosystems have been introduced by von Ahn and Hopper [11] and by Backes and Cachin [2]. The latter work also goes beyond the case of passive adversaries considered here and models adaptive chosen-coverttext attacks, which are similar to adaptive chosen-ciphertext attacks against public-key cryptosystems. Achieving security against such attacks results in the strongest security notion known today for public-key cryptosystems and for public-key stegosystems.

As this brief survey of steganography shows, the evolution of the formal approach to stegosystems has gone through the same steps as the development of formal models for cryptosystems. The models and the formulation of corresponding stegosystems that offer provable security have greatly enhanced our understanding of this important area of information security.

References

- [1] R. J. Anderson and F. A. Petitcolas, “On the limits of steganography,” *IEEE Journal on Selected Areas in Communications*, vol. 16, May 1998.
- [2] M. Backes and C. Cachin, “Public-key steganography with active attacks,” in *Proc. 2nd Theory of Cryptography Conference (TCC 2005)* (J. Kilian, ed.), vol. 3378 of *Lecture Notes in Computer Science*, pp. 210–226, Springer, 2005.
- [3] C. Cachin, “An information-theoretic model for steganography,” *Information and Computation*, vol. 192, pp. 41–56, July 2004. (Preliminary version appeared in *Proc. 2nd Workshop on Information Hiding*, *Lecture Notes in Computer Science*, vol. 1525, Springer, 1998.).
- [4] O. Goldreich, *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.
- [5] 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*, pp. 77–92, Springer, 2002.
- [6] S. Katzenbeisser and F. A. P. Petitcolas, “Defining security in steganographic systems,” in *Security and Watermarking of Multimedia Contents IV* (E. J. Delp and P. W. Won, eds.), vol. 4675 of *Proceedings of SPIE*, pp. 260–268, International Society for Optical Engineering, 2002.
- [7] M. Naor and A. Shamir, “Visual cryptography,” in *Advances in Cryptology: EUROCRYPT ’94* (A. De Santis, ed.), vol. 950 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, 1995.
- [8] B. Pfitzmann, “Information hiding terminology,” in *Information Hiding, First International Workshop* (R. Anderson, ed.), vol. 1174 of *Lecture Notes in Computer Science*, pp. 347–350, Springer, 1996.
- [9] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, pp. 656–715, Oct. 1949.
- [10] 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.
- [11] L. von Ahn and N. J. Hopper, “Public-key steganography,” in *Advances in Cryptology: Eurocrypt 2004* (C. Cachin and J. Camenisch, eds.), vol. 3027 of *Lecture Notes in Computer Science*, pp. 322–339, Springer, 2004.