Efficient Provably Secure Public Key Steganography

Tri Van Le [∞]

Department of Computer Science
Florida State University
Tallahassee, Florida 32306-4530, USA.
Email: levan@cs.fsu.edu

Abstract. We construct *efficient* public key steganographic schemes, without resort to any special existence assumption such as unbiased functions. This is the first time such a construction is obtained. Not only our constructions are *secure*, but also are essentially optimal and have *no error* decoding. We achieve this by designing a new primitive called $\mathcal{P}$-codes.

Keywords: foundation, steganography, public key, computational security, coding theory.

1 Introduction

Motivations. The *Prisoner's Problem* introduced by G.J. Simmons [13] and generalized by R. Anderson [1] can be stated informally as follows: Two prisoners, Alice and Bob, want to communicate to each other their secret escape plan under the surveillance of a warden, Wendy. In order to pass Wendy's censorship, Alice and Bob have to keep their communications as innocent as possible so that they will not be banned by Wendy.

Existing results. Previously, the Prisoner's Problem was considered in the secret key setting by: Cachin [2], Mittelholzer [10], Moulin and Sullivan [11], Zollner et.al. [14] in the unconditional security model; and Katzenbeisser and Petitcolas [9], Hopper et.al. [7], Reyzin and Russell [12] in the conditional security model. In this article, we consider the problem in the *public key* setting. In this setting, Craver [3] and Anderson[1] proposed several heuristic ideas to solve the problem. Katzenbeisser and Petitcolas [9] gave a formal model. Hopper and Ahn [8] constructed proven secure schemes assuming the existence of *unbiased functions* (more details on page 2). Unbiased functions are currently required for *all public key schemes* in the literature [8], and are the main ingredients in most of other secure generic steganographic schemes [2, 7, 12]. Further, current approaches [2, 7, 8, 12] result in very low information rate for steganography in both the public key and secret key settings. We show here an alternative paradigm to securing public key steganography without using unbiased functions. Our approach also improves the efficiency of existing schemes.

Purpose. The contributions of this article are the following:

- A novel general paradigm for designing secure steganographic systems.
- Highly efficient rate for both *public key* and *private key* steganographic systems.
- No error in encoding and decoding operations.
- Public key steganography without dependence on unbiased functions.

[∞] This work was supported by NSF grant 9903216.

Organization. The article is organized as follows: we describe the model in Section 2, our new primitive $\mathcal{P}$-codes in Section 3, show constructions of public key steganographic schemes and their security proofs in Section 4, and give a rate calculation for our schemes in Section 5. We conclude in Section 6.

1.1 Informal Discussions

Model. Generally speaking, a public key steganographic system consists of three randomized algorithms: a `Setup` algorithm that creates a public-private key pair, an `Embed` algorithm that creates a stegotext from a given pair of a hiddentext and a public key, and an `Extract` algorithm that extracts the embedded hiddentext from given stegotext using a secret key. Besides the trivial relationship between the `Embed` and `Extract` algorithms, the `Embed` algorithm must produce stegotexts in such a way that they evade the suspicion of Wendy. This requires that the probability distribution of the stegotexts is indistinguishable from the distribution of innocent messages or covertexts. In practice, it is hard to know the distribution of innocent messages, be they ordinary English texts, audio, video, or still images. Therefore we assume that only a message sampler, which can produce innocent covertexts, is given. A complication in reality is that successive messages may be dependent on each other. So we allow the sampler to be stateful, i.e. it may have internal memory to remember its previously generated messages [9, 7]. The objective for Alice is then to produce stegotext sequences that are indistinguishable from the covertext streams produced by this sampler.

Observations. The only difference between steganographic schemes and encryption schemes is that the former ones must produce output indistinguishable from the covertext distributions. This is the crucial point that distinguishes steganography from normal cryptography. In standard cryptography, almost all messages or ciphertexts are uniformly randomly distributed. Whereas in the case of steganography, the distribution of the stegotexts, the equivalence of ciphertexts, are pre-specified. Our next observation is the following.

If we assume that there are public invertible transformations that transform uniform random strings to/from non-uniform covertexts, then all problems in steganography could be solved modularly: first we apply cryptography to produce protocols with uniformly random messages; then these messages are transformed into non-uniform covertexts at the sender's side, and later transformed back into uniformly random messages at the receiver's side.

It is expected that all normal cryptographic properties of our protocols would be preserved by these public transformation. Unbiased functions, first appeared in [1] as a possible way for designing steganography, are a very special case of this. They are publicly known functions that translate a randomly chosen covertext string into a uniformly random bit as in the following definition.

Definition 1. *A function $f : C \to \{0, 1\}$ is called* unbiased *with respect to a covertext distribution $\mathcal{P}$ over a covertext space C if:*

$$bias(f, \mathcal{P}) = \left| \Pr_{c \in_{\mathcal{P}} C}[f(c) = 0] - \frac{1}{2} \right|$$

is negligible in the security parameter t, where $c \in_{\mathcal{P}} C$ means c is a covertext chosen randomly from C accordingly to the covertext distribution $\mathcal{P}$.

The corresponding backward transformation $f^{-1}: \{0,1\} \to C$ is trivial using exhaustive search, i.e. sample covertexts until one with proper f-value appears.

Note that these functions likely do not exist in many of cases, especially when the covertext space C is not very large, for example when $|C| = 2$. Of course then one can construct these functions on compounded covertext space C^n instead ($n > 1$) as done in [8]. However, this approach results in extremely low information rates, even with additional improvement of [12].

Our Solution. We solve the steganographic problem in a novel way. At the heart of our solution are *uniquely decodable variable length coding schemes Γ, called $\mathcal{P}$-codes, with source alphabet Σ and destination alphabet C such that: if $x \in \Sigma^\infty$ is chosen uniformly randomly then $\Gamma(x) \in C^\infty$ distributes according to $\mathcal{P}$, where $\mathcal{P}$ is a given distribution over sequences of covertexts.*

Note that such a coding scheme is quite related to homophonic coding schemes [6], which are uniquely decodable variable length coding scheme Γ' with source alphabet C and destination alphabet Σ such that: if $c \in C^*$ is chosen randomly according to distribution $\mathcal{P}$ then $\Gamma'(c) \in \Sigma^*$ is a sequence of independent and uniformly random bits.

Of course, one can hope that such a homophonic coding scheme Γ' will give rise to a uniquely decodable $\mathcal{P}$-code Γ. However, this is not necessarily true because Γ' can map one-to-many, as in the case of [6]. Therefore by exchanging the encoding and decoding operations in Γ', we will obtain a non-uniquely decodable $\mathcal{P}$-coding scheme Γ'', which is not what we need.

To construct these $\mathcal{P}$-codes, we generalize a heuristic idea of Ross Anderson [1] where one can use a perfect compression scheme on the covertexts to obtain a perfectly secure steganographic scheme. Nevertheless, in practice one can never obtain a perfect encryption scheme, so we have to build our $\mathcal{P}$-coding schemes based on non-perfect compression schemes, such as arithmetic compression. The result is a coding scheme which achieve near optimal information rate, and has no error.

2 Definitions

2.1 Channel

Let C be a finite *message space*. A *channel* $\mathcal{P}$ is a probability distribution over the space C^∞ of infinite message sequences $\{(c_1, c_2, \dots) \mid c_i \in C, i \in \mathbb{N}\}$. The communication channel $\mathcal{P}$ may be *stateful*. This means that: for all $n > 0$, c_n might depend probabilistically on $c_1, \dots, c_{n-1}$. When individual messages are used to embed hiddentexts, they are called covertexts. Therefore C is also called the covertext space. Denote C^* the space of all finite message sequences $\{(c_1, \dots, c_l) \mid l \in \mathbb{N}, c_i \in C, 1 \le i \le l\}$. If $h \in C^*$ is a prefix of $s \in C^\infty$, that is $s_i = h_i$ for all $1 \le i < \ell(h)$, then we write $h \subset s$. The expression $s \in_{\mathcal{P}} C^\infty$ means that s is chosen randomly from C^∞ according to distribution $\mathcal{P}$. Denote $\mathcal{P}(c) = \Pr[c \subset s \mid s \in_{\mathcal{P}} C^\infty]$ for all $c \in C^*$.

Sampler. A *sampler* S for the channel $\mathcal{P}$ is a sampling oracle such that upon a query $h \in C^*$, S randomly outputs a message $c_i \in C$ according to the marginal probability distribution $\mathcal{P}_h$:

$$\mathcal{P}_h(c_i) = \Pr[(h \| c_i) \subset s \mid h \subset s \ \wedge \ s \in_{\mathcal{P}} C^\infty],$$

where $h \| c_i$ is the concatenation of h and c_i. In general, we define $\mathcal{P}_h(c) = \Pr[(h \| c) \subset s \mid h \subset s \ \wedge \ s \in_{\mathcal{P}} C^\infty]$ for all $h \in C^*$ and $c \in C^* \cup C^\infty$. The expression $s = S(h)$ means s is the result of querying $S(h)$. Since S responses randomly, each individual query may have a different result. Finally, $x \in_R X$ means x is chosen uniformly randomly from the set X. Finite messages sequences can always be included in $\mathcal{P}$ by appending copies of a special *null* symbol to get infinite sequences.

Assumption. From now on, we assume that $\mathcal{P}$ is a channel over message space C, and that a corresponding sampler S is given. The channel $\mathcal{P}$ represents the probability distribution of an innocent communication channel; the sampler S generates covertexts according to $\mathcal{P}$, see [1, 7, 9]. Our purpose is to construct steganographic systems whose stegotext distributions are indistinguishable from $\mathcal{P}$. We also assume that the query h given to sampler S is always the history of messages communicated between Alice and Bob.

2.2 Steganographic systems

A public key *steganographic system* is specified by a pair of key spaces $\mathcal{K}_e \times \mathcal{K}_d$, and three randomized algorithms, `Setup`, `Embed`, `Extract`, that works as follows:

- `Setup`: takes a security parameter k as input, and returns system parameters `params` and a pair of keys $(e, d) \in \mathcal{K}_e \times \mathcal{K}_d$. Among other things, the system parameters `params` include a short description of a finite hiddentext space $\mathcal{M}$.
- `Embed`: takes as input a public key $e \in \mathcal{K}_e$, a hiddentext $m \in \mathcal{M}$, and returns a stegotext $s \in C$. The algorithm may query the sampler S.
- `Extract`: takes as input a secret key $d \in \mathcal{K}_d$, a stegotext $s \in C$, and returns either the symbol $\perp$ on failure, or a hiddentext $m \in \mathcal{M}$.

As usual, we require that `Extract`$(d, \cdot)$ reverses the action of `Embed`$(e, \cdot)$.

2.3 Security Objectives

Chosen hiddentext security. The task of warden Wendy is to distinguish two cases: whether the communications between the prisoners are innocent, or contain hiddentexts. In order to detect hiddentexts, Wendy is allowed to mount chosen hiddentext attacks, which are plausible in practice when Wendy has oracle access to the embedding machine and would like to detect the use of this machine to communicate steganographically.

Chosen hiddentext attacks on steganographic systems are parallel to chosen plaintext attacks on encryption systems. The only difference is in the purposes of the two attacks. In the first attack, the objective is to detect the existence of hidden messages or hiddentexts. In the second attack, the objective is to discover partial information about the content of the secret messages. Our definition of *chosen hiddentext security* reflects this difference:

- In an indistinguishability under chosen plaintext attack (IND-CPA), the challenger randomly chooses one of the two plaintexts submitted by the adversary and encrypts it. An encryption scheme is secure against this attack if an adversary cannot tell which plaintext was encrypted.
- In a hiding under chosen hiddentext attack (HID-CSA), the challenger randomly flips a coin, and depending on the result decides to encrypt the submitted hiddentext or to randomly sample a cover message. A steganographic scheme is secure against this attack if an adversary cannot tell stegotexts from covertexts.

While the hiding objective of steganographic systems is substantially different from the semantic security objective of encryption systems, it is not hard to see that HID-CHA security implies IND-CPA, as shown in [7] in the secret key setting.

Formally, we say that a steganographic system is secure against an chosen hiddentext attack if no polynomial time adversary $\mathcal{W}$ has non-negligible advantages against the challenger in the following game:

- Setup: The challenger takes a security parameter k and runs `Setup` algorithm. It gives the resulting system parameters `params` to the adversary, and keeps the keys (e, d) to itself. In the case of public key system, the adversary also gets e included with system parameters `params`.
- Warmup: The adversary issues j queries $m_1, \ldots, m_j$ where each query m_i is a hiddentext in $\mathcal{M}$. The challenger responds to each query m_i by first running `Embed` algorithm with input key e and message m_i, then sending the corresponding result of $\texttt{Embed}(e, m_i)$ back to the adversary. The queries may be chosen adaptively by the adversary.
- Challenge: The adversary stops Phase 1 when it desires, and sends a hiddentext $m \in \mathcal{M}$ to the challenger. The challenger then picks a random bit $b \in \{0, 1\}$ and does the following:
  - If $b = 0$, the challenger queries S for a covertext $s = S(h)$, and sends s back to the adversary.
  - If $b = 1$, the challenger runs the `Embed` algorithm on key e and hiddentext m, and sends the resulting stegotext $s = \texttt{Embed}(e, m)$ back to the adversary.
- Guess: The adversary outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b' = b$.

Such an adversary $\mathcal{W}$ is called an HID-CHA attacker. We define the adversary $\mathcal{W}$'s advantage in attacking the system as $|\Pr[b' = b] - \frac{1}{2}|$ where the probability is over the random coin tosses of both the challenger and the adversary.

We remind you that a standard IND-CPA attacker would play a different game, where at the challenge step:

- Challenge: The adversary sends a pair of plaintexts $m_0, m_1 \in \mathcal{M}$ upon which it wishes to be challenged to the challenger. The challenger then picks a random bit $b \in \{0, 1\}$, runs the encryption algorithm on public key e and plaintext m_b, and sends the resulting ciphertext $c = \texttt{Encrypt}(e, m_b)$ back to the adversary.

As in IND-CPA game against an encryption system, we also define an IND-CHA game against a steganographic system. The definition is exactly the same, except with necessary changes of names: the `Encrypt` and `Decrypt` algorithms are replaced by the `Embed` and `Extract` algorithms; and the terms plaintext and ciphertext are replaced by the terms hiddentext and stegotext, respectively. Similarly, a steganographic system is called IND-CPA secure if every polynomial time adversary $\mathcal{W}$ has negligible advantages in an IND-CPA game against the steganographic system.

3 Construction of $\mathcal{P}$-Codes

A uniquely decodable coding scheme Γ is a pair consisting of a probabilistic encoding algorithm Γ_e and a deterministic decoding algorithm Γ_d such that $\forall m \in dom(\Gamma_e) : \Gamma_d(\Gamma_e(m)) = m$. In this article, we are interested in coding schemes whose source alphabet is binary, $\Sigma = \{0, 1\}$.

Definition 2. *Let $\mathcal{P}$ be a channel with message space C. A $\mathcal{P}$-code, or a $\mathcal{P}$-coding scheme, is a uniquely decodable coding scheme Γ whose encoding function $\Gamma_e : \Sigma^* \to C^*$ satisfies:*

$$\epsilon(n) = \sum_{c \in \Gamma_e(\Sigma^n)} \left| \Pr\left[\Gamma_e(x) = c \mid x \in_R \Sigma^n\right] - \mathcal{P}(c) \right|$$

is a negligible function in n. In other words, the distribution of $\Gamma_e(x)$ is statistically indistinguishable from $\mathcal{P}$ when x is chosen uniformly randomly. The function

$$e(n) = \frac{1}{n} \sum_{c \in \Gamma_e(\Sigma^n)} \mathcal{P}(c) H_{\mathcal{P}}(c)$$

is called the expansion rate of the encoding. [1]

Let $\mathcal{P}$ be a channel with sampler S. We assume here that $\mathcal{P}_h$ is polynomially sampleable, which was also assumed in [7, 8] in order to achieve proven security.[2] This is equivalent to saying that S is an efficient algorithm that given a sequence of covertexts $h = (c_1, \ldots, c_n)$ and a uniform random string $r \in_R \{0,1\}^{R_n}$, S outputs a covertext $c_{n+1} \in C$ accordingly to probability distribution $\mathcal{P}_h$. Nevertheless, we assume less that the output of S to be statistically close to $\mathcal{P}_h$.

We use algorithm S to construct a $\mathcal{P}$-coding scheme Γ. For $x = (x_1, \ldots, x_n) \in \Sigma^n$, denote $\overline{x}$ the non-negative integer number whose binary representation is x. For $0 \le a \le 2^n$, denote $\underline{a} = (a_1, \ldots, a_n)$ the binary representation of integer number a. In the following, let t be an integer parameter, h_0 is the history of all previous communicated messages between Alice and Bob. Further let us assume that the distribution $\mathcal{P}_h$ has minimum entropy bounded from below by a constant $\xi > 0$. Let $H : C^{l_0} \times \mathbb{N} \to \{0,1\}^{R_n}$ be a cryptographically secure family of pseudo random functions (secretly shared between sender and receiver), where $l_0 \ge k/\xi$ and k is the security parameter, such that $H(X, \cdot)$ is pseudo random even when $X \in C^{l_0}$ is chosen accordingly to $\mathcal{P}$. This can be generally achieved in practice using secure hash functions and pseudo random generators. Let U_n be a uniform random variable over $\{0,1\}^{R_n}$.

Γ_1-***Encode***. **Input**: $x = (x_1, \ldots, x_n) \in \Sigma^n$.
Output: $c = (c_1, \ldots, c_l) \in C^*$.

1. **let** $a = 0, b = 2^{2n}, h = \epsilon$.
2. **let** $c'_i = S(h_0 \| h, U_n)$ for $0 \le i < l_0$.
3. **let** $z = (c'_0, \ldots, c'_{l_0 - 1})$.

---

[1] Ideally, we would have used $\Pr\left[\Gamma_e(x) = c \mid x \in_R \Sigma^n\right]$ instead of $H_{\mathcal{P}}(c)$. However, the two distributions are statistically indistinguishable so this makes no real difference.

[2] Theoretically, allowing $\mathcal{P}_h$ to be non-polynomially sampleable would allow hard problems to be solvable.

4. **while** $\lceil a/2^n \rceil < \lfloor b/2^n \rfloor$ **do**
   (a) **let** $c_i^* = S(h_0 \| z \| h, H(z, t \cdot len(h) + i))$ for $0 \leq i < t$.
   (b) Order the c_i^*'s in some fixed increasing order:
   $$c_0^* = \cdots = c_{i_1-1}^* < c_{i_1}^* = \cdots = c_{i_2-1}^* < \cdots < c_{i_{m-1}}^* = \cdots = c_{t-1}^*,$$
   where $0 = i_0 < i_1 < \cdots < i_m = t$.
   (c) **let** $0 \leq j \leq m-1$ be the unique j such that
   $$i_j \leq \lfloor (2^n \overline{x} - a)t/(b-a) \rfloor < i_{j+1}.$$
   (d) **let** $a' = a + (b-a)i_j/t$, $b' = a + (b-a)i_{j+1}/t$.
   (e) **let** $(a, b) = (a', b')$.
   (f) **let** $h = h | c_{i_j}^*$.
5. Output $c = z \| h$.

Everyone who is familiar with information theory will immediately realize that the above encoding resembles to the arithmetic decoding of number $\overline{x}/2^n$. In fact, the arithmetic encoding of the sequence c is exactly the number $\overline{x}/2^n$.

Each time the sender outputs a covertext $c_{i_j}^*$, the receiver will obtain some information about the message x, i.e. the receiver is able to narrow the range $[a, b]$ containing $2^n x$. The sender stops sending more covertexts until the receiver can completely determine the original value x, i.e. when the range $[a, b]$ is less than 2^n. Thus the decoding operation for the $\mathcal{P}$-coding scheme Γ follows.

Γ_1-`Decode`. **Input**: $c = (c_1, \ldots, c_l) \in C^*$.
Output: $x = (x_1, \ldots, x_n) \in \Sigma^n$.

1. **let** $a = 0, b = 2^{2n}, h = \epsilon$.
2. **let** $c_i' = c_{i+1}$ for $0 \leq i < l_0$.
3. **let** $z = (c_0', \ldots, c_{l_0-1}')$.
4. **for** ind **from** 0 **to** $|c| - l_0 - 1$ **do**
   (a) **let** $c_i^* = S(h_0 \| z \| h, H(z, t \cdot ind + i))$ for $0 \leq i \leq t-1$.
   (b) Order the c_i^*'s in some fixed increasing order:
   $$c_0^* = \cdots = c_{i_1-1}^* < c_{i_1}^* = \cdots = c_{i_2-1}^* < \cdots < c_{i_{m-1}}^* = \cdots = c_{t-1}^*,$$
   where $0 = i_0 < i_1 < \cdots < i_m = t$.
   (c) **let** $0 \leq j \leq m-1$ be the unique j such that $c_{i_j}^* = c_{(l_0+1+ind)}$.
   (d) **let** $a' = a + (b-a)i_j/t$, $b' = a + (b-a)i_{j+1}/t$.
   (e) **let** $(a, b) = (a', b')$.
   (f) **let** $h = h | c_{i_j}^*$.
5. **let** $v = \lceil a/2^n \rceil$.
6. Output $x = \underline{v}$.

If x is chosen uniformly randomly from Σ^n then the correctness of our $\mathcal{P}$-coding scheme Γ is established through the following theorem.

Theorem 1. *Γ_1 described above is a $\mathcal{P}$-code.*

Proof. First, z is transmitted in plain so in each iteration, the encoding and decoding operations use the same list of $c_0^*, \ldots, c_{t-1}^*$. Therefore, the values of $i_0, \ldots, i_t, j, a', b', h, a, b$ in the encoding are the same as in the decoding. Further, due to our choice of j, $2^n\overline{x} \in [a, b)$ is true not only before the iterations, but also after each iteration. Therefore at the end of the encoding, we obtain $\lceil a2^{-n} \rceil = \lfloor b2^{-n} \rfloor = \overline{x}$. Because the values of a, b in encoding are the same as in decoding, this shows that the decoding operation's output is the same as the encoding operation's input x, i.e. $\mathit{\Gamma}_1$ is uniquely decodable. Next, we will prove that it is also a $\mathcal{P}$-code.

Indeed by definition, at each corresponding iteration $H(z, t \cdot len(h) + i) = H(z, t \cdot ind + i)$ is indistinguishable from uniformly random. Now assume temporarily that a, b were real numbers. Note that the covertexts $c_0^*, \ldots, c_{t-1}^*$ are generated independently of x, so $i_0, \ldots, i_t$ are also independent of x. By simple induction we can see that after each iteration $i \leq l - 1$, the conditional probability distribution of $\overline{x}$ given the history $h = c_1 \| \ldots \| c_i$, is uniformly random over integers in the range $[a2^{-n}, b2^{-n})$. However, in our algorithms the numbers a, b are represented as integers using rounding. So the conditional distribution of $\overline{x}$ at the end of each iteration except the last one is not uniformly random, but anyway at most $4/(b-a) \leq 2^{2-n}$ from uniformly random due to rounding, and due to the fact that $b - a \geq 2^n$. Since 2^{2-n} is negligible, and our encoding operations are polynomial time, they can not distinguish a truly uniformly random $\overline{x}$ from a statistically-negligible different one. So for our analysis, we can safely assume that $\overline{x}$ is indeed uniformly random in the range $[a2^{-n}, b2^{-n})$ at the *beginning* of each iteration, including the last one.

Then at the beginning of each iteration i, conditioned on the previous history $h = c_0 \| \ldots \| c_{i-1}$, $u = \lfloor (2^n\overline{x} - a)t/(b - a) \rfloor$ is a uniformly random variable on the range $[0, t - 1]$, thus u is probabilistically independent of $c_0^*, \ldots, c_{t-1}^*$. Since $c_0^*, \ldots, c_{t-1}^*$ are identically distributed, c_u must also be distributed identically. Further, by definition, $i_j \leq k < i_{j+1}$, so $c_u = c_{i_j}^* = c_i$. Hence c_i distributes identically as each of $c_0^*, \ldots, c_{t-1}^*$ does. By definition of S, this distribution is $\mathcal{P}_{h_0 \| h}$, i.e. c distributes accordingly to $\mathcal{P}_{h_0}$. Since x is not truly uniformly random but rather statistically indistinguishable from uniformly random, we conclude that the output c of the encoding operation is statistically indistinguishable from $\mathcal{P}_{h_0}$. Therefore, by definition, our coding scheme is indeed a $\mathcal{P}$-code. Our coding scheme has a small overhead rate of $l_0/n = k/n\xi$. However, this overhead goes to 0 when $n > k^{1+\epsilon}$ as $n \to \infty$ and $\epsilon > 0$. Therefore our encoding is essentially optimal. See our formal proof in Section 5.

Note that in the case that $m = 0$, the encoding/decoding operations still work correctly, i.e. there are no errors. In such case, the range $[a, b)$ does not change: the encoding will output c_0^* without actually embedding any hidden information, while the decoding operation will read c_0^* without actually extracting any hidden information. This happens more often when the entropy of the cover distribution is very near zero. However, from now on we will assume that our distribution $\mathcal{P}_h$ will have minimal entropy bounded from below by a fixed constant $1 > \rho > 0$, i.e. $\forall h \in C^*, c \in C : \mathcal{P}_h(c) < \rho$. Then with overwhelming probability of at least $1 - |C|\rho^t$, we will have $m > 0$.

4 Construction of Steganographic Systems

Our purpose in this section is to construct steganographic systems based on the $\mathcal{P}$-coding scheme $\mathit{\Gamma}$. Using the notations from Sections 2 and 3, our construction is the following. Here, h denotes the history of previously communicated messages.

4.1 Private Key Steganographic Systems

Let G be a cryptographically secure pseudo-random generator, and k be a shared secret key. In the setup step, k is given as seed to G.

S_1-***Embed.*** **Input**: $m \in \Sigma^n$.
Output: $c \in C^*$.

1. **let** $r \in_R \Sigma^n$ be the next n random bits generated by G.
2. Output $c = \Gamma_e(r \oplus m)$.

S_1-***Extract.*** **Input**: $c \in C^*$.
Output: $m \in \Sigma^n$.

1. **let** $r \in_R \Sigma^n$ be the next n random bits generated by G.
2. Output $m = \Gamma_d(c) \oplus r$.

Theorem 2. *The steganographic scheme S_1 is CHA-secure.*

Proof. The proof is straight-forward: $r \oplus m$ is computationally indistinguishable from uniformly random, so by the property of Γ_e, the output covertext sequence $c = \Gamma_e(r \oplus m)$ is computationally indistinguishable from $\mathcal{P}$. Further, each time the embedding operation is performed, the pseudo-random generator G changes its internal state, so its output r are independent of each others in the attacker's view. Consequently, the values of $r \oplus m$, and so do the values of $c = \Gamma_e(r \oplus m)$, are probabilistically independent of each others to the attacker. This means that the ciphertexts obtained by the attacker in the warmup step do not help him in the guessing step in anyway. Therefore our scheme is secure against chosen hiddentexts attack.

Expansion Rate. It is clear that the expansion rate of this scheme is the same as the expansion rate of the $\mathcal{P}$-code. Additionally, both sides must maintain the status of the generator G. However, this status is very small, similar to a synchronized counter used in [7]. Note that in this private-key case, we can embed a little bit more efficient by not using the preamble z in the encoding/decoding operations because both sides already share a secret pseudo-random tape.

4.2 Public Key Steganographic Systems

In this section, we use the idea of Diffie-Hellman key exchange to obtain an efficient public key steganographic scheme. Denote $H_{\mathcal{P}}(c) = -\log_2(\mathcal{P}(c))$ the entropy of $c \in C^*$ according to the covertext distribution $\mathcal{P}$. We assume that there exists a constant $0 < \rho < 1$ such that:

$$\forall h \in C^*, \forall c \in C : \mathcal{P}_h(c) < \rho.$$

In other words, $\mathcal{P}_h$ has its minimum entropy bounded from below by a positive constant $(-\log_2(\rho))$.

S_2-***Setup.*** The system parameter is a generator g of a prime order cyclic group $<g>$, whose decisional Diffie-Hellman problem is hard. Let (g, g^a) be the public key of sender Alice, and (g, g^b) be the public key of receiver Bob. Let $F(X, Y)$ be a public cryptographically secure family of pseudo-random functions, indexed by variable $X \in <g>$. Let k be the security parameter and $n = O(poly(k))$. The embedding and extracting operations are as follows.

S_2-***Embed.*** **Input**: $m \in \{0,1\}^n$.
Output: $c \in C^*$.

1. Let $l = \lceil \frac{k}{\log_2 \frac{1}{\rho}} \rceil$, $h_0 = \epsilon$.
2. **for** i **from** 1 **to** l **do** $c_i = S(h_0)$; $h_0 = h_0 \| c_i$.
3. Let $r = F((g^b)^a, h_0)$.
4. Output $c = h_0 \| \Gamma_e(r \oplus m)$.

Note that in the call to $\Gamma_e(r \oplus m)$, we initialize h with h_0, instead of ϵ.

S_2-***Extract.*** **Input**: $c \in C^*$.
Output: $m \in \{0,1\}^n$.

1. Let $l = \lceil \frac{k}{\log_2 \frac{1}{\rho}} \rceil$, $c = (h_0, c')$ where $|h_0| = l$.
2. Let $r = F((g^a)^b, h_0)$.
3. Output $m = \Gamma_d(c') \oplus r$.

Note that we initialize h with h_0 instead of ϵ in the call to $\Gamma_d(c')$. Similarly to the construction $F((g^a)^b, \cdot)$, the secretly shared family of pseudo random function H used in Γ_e, Γ_d can be constructed from a public family H' with index g^{ab}, e.g. using $H(X,Y) = H'(g^{ab}, X, Y)$.

Theorem 3. *The steganographic scheme S_2 is CHA-secure.*

Proof. By definition of the family F and the hardness of the Diffie-Hellman problem over $<g>$, we obtain that g^{ab}, and therefore r, is computationally indistinguishable from uniformly random. Thus, by definition of our $\mathcal{P}$-code, c is computationally indistinguishable from $\mathcal{P}$.

Further, since $H_{\mathcal{P}}(h_0) \geq k$, with overwhelming probability h_0 is different each time we embed. Therefore even when the embedding oracle is queried repeatedly, r still appears to the attacker as independently and uniformly random. Therefore in the attacker's view the ciphertexts obtained by him in the warmup step are independent of of the challenged ciphertext, i.e. they are useless for the attack. That means our scheme is CHA-secure.

Expansion Rate. The expansion rate of this scheme equals to the rate of the underlying $\mathcal{P}$-code plus the overhead in sending h_0. Nevertheless, the overhead of h_0, which is $O(\lceil \frac{k}{\log_2(\frac{1}{\rho})} \rceil)$, only depends on the security parameter k. Thus it diminishes when we choose n large enough so that $k = o(n)$, say $n = k \log(k)$. Therefore the expansion rate of our steganographic system is essentially that of the $\mathcal{P}$-code.

5 Essentially Optimal Rates

In this section we consider applications of our schemes in two cases: distribution $\mathcal{P}$ is given explicitly by a cumulative distribution function F, and is given implicitly by a black-box sampler S. In both cases, we show that the achieved information rate is essentially optimal.

5.1 Cumulative Distribution Function

We show here that in case we have additionally a cumulative distribution function F of the given distribution, then the construction can be much more efficient. First, let us define what a cumulative distribution function is, and then how to use this additional information to construct $\mathcal{P}$-coding schemes.

Let the message space C be ordered in some strict total order $'<'$ so that $v_0 < v_1 < \dots$ is a sorted sequence of all covertexts. A *cumulative distribution function* (CDF) for the channel $\mathcal{P}$ is a family of functions $F_h : C \to [0, 1]$ such that $F_h(v) = \sum_{v'<v} \mathcal{P}_h(v')$ for all $h \in C^*$ and $v \in C$. We modify our $\mathcal{P}$-code slightly so that it can use the additional information available effectively.

Γ_2`-Encode`. **Input**: $x = (x_1, \dots, x_n) \in \Sigma^n$.
Output: $c = (c_1, \dots, c_l) \in C^*$.

1. **let** $a = 0, b = 2^{2n}, h = \epsilon$.
2. **while** $\lceil a/2^n \rceil < \lfloor b/2^n \rfloor$ **do**
   (a) **let** $i_j = tF_{h_0\|h}(v_j)$.
   (b) **let** j be the unique integer such that
   $$i_j \leq \lfloor (2^n \overline{x} - a)t/(b - a) \rfloor < i_{j+1}.$$
   (c) **let** $a' = a + (b - a)i_j/t$, $b' = a + (b - a)i_{j+1}/t$.
   (d) **let** $(a, b) = (a', b')$.
   (e) **let** $h = h|v_{i_j}$.
3. Output $c = h$.

The only difference here is that instead of calling S repeatedly to generate c_i^* $(0 \leq i \leq t - 1)$ and then deduce i_j $(0 \leq j \leq m - 1)$, we use $v_0, v_1, \dots \in C$ directly and let $i_j = tF_{h_0\|h}(c_j)$ for $j = 0, 1, \dots$. Note that the sorted sequence $v_0, v_1, \dots$ of all covertexts can either be given explicitly, or be given by a function $v : \mathbb{N} \to C$. In the either case, the determination of j in step $2(b)$ can be done by binary searching, thus allows large covertext space C to be used.

Γ_2`-Decode`. **Input**: $c = (c_1, \dots, c_l) \in C^*$.
Output: $x = (x_1, \dots, x_n) \in \Sigma^n$.

1. **let** $a = 0, b = 2^{2n}, h = \epsilon$.
2. **for** i **from** 1 **to** $|c|$ **do**
   (a) **let** $i_j = tF_{h_0\|h}(v_j)$.
   (b) **let** j be the unique integer such that $v_{i_j} = c_i$.
   (c) **let** $a' = a + (b - a)i_j/t$, $b' = a + (b - a)i_{j+1}/t$.
   (d) **let** $(a, b) = (a', b')$.
   (e) **let** $h = h|v_{i_j}$.
3. **let** $v = \lceil a/2^n \rceil$.
4. Output $x = \underline{v}$.

Theorem 4. *The coding scheme described above is a $\mathcal{P}$-code.*

Proof. The proof is the same, word by word, as in proof of Theorem 1, with only necessary changes of c_i^* and i_j as noted above.

Theorem 5. *The expansion rate $e(n)$ is bounded from above by $1 + \frac{1}{n}\log_2(|C|)$.*

Proof. At each iteration i, the range $[a, b]$ is reduced in size by a factor of $(b' - a')/(b - a) = (i_{j+1} - i_j)/t = F_h(v_{j+1}) - F_h(v_j) = \mathcal{P}_h(v_{i_j}) = \mathcal{P}_h(c_i)$. Further, before the last iteration $b - a \geq 2^n$, so we get:

$$\mathcal{P}(c_1 \| \dots \| c_{l-1}) = \prod_{i=1}^{l-1} \mathcal{P}_{c_1 \| \dots \| c_{i-1}}(c_i) \geq \frac{2^n}{2^{2n}} = 2^{-n}.$$

This means $H_{\mathcal{P}}(c_1 \| \dots \| c_{l-1}) \leq n$. Summing over all $x \in \Sigma^n$ we get:

$$\sum_{c \in \Gamma_e(\Sigma^n)} \mathcal{P}(c) H_P(c) \leq n + \log_2(|C|).$$

This shows that the expansion rate $e(n)$ is bounded above by:

$$e(n) = \frac{1}{n} \sum_{c \in \Gamma_e(\Sigma^n)} \mathcal{P}(c) H_P(c) \leq 1 + \frac{\log_2(|C|)}{n}.$$

Since $\log_2(|C|)$ is a constant, we obtain that $e(n) \to 1$ when $n \to \infty$.

5.2 General Case

In this case, we know nothing about the distribution $\mathcal{P}_h$, except a given black box sampler S. We give a proof showing that our scheme is optimal.

Theorem 6. *The $\mathcal{P}$-code defined in Section 3 is essentially optimal.*

Proof. First, note that any steganographic scheme defined over channel $\mathcal{P}$ is indeed a $\mathcal{P}$-coding scheme. Second, the expansion rate of our steganographic schemes is essentially the expansion rate of the underlying $\mathcal{P}$-code. Hence it is enough to show that our $\mathcal{P}$-code Γ_1 is optimal.

Indeed, let Γ' be any $\mathcal{P}$-coding scheme that works generically like Γ, i.e. Γ' works on any black box S whose output has minimal entropy bounded from below (e.g. by ξ). Let t be the number of oracle calls to S by Γ', and let $c^* = (c_0^*, \dots, c_{t-1}^*)$ be the corresponding results.

Then Γ' can only return one of the covers $c_0^*, \dots, c_{t-1}^*$ as its next stegotext to be sent to the receiver. Indeed, assume otherwise that this is not the case. Then consider a black box sampler S' that output covertexts including a long random string signed with a secure digital signature. Now apply Γ' to S'. If Γ' outputs anything that is not in the list of covers returned by S', the output of Γ' will not contain a valid signature, or otherwise by definition the digital signature would have been insecure. Now such unsigned covers is immediately detectable by a polynomial time algorithm, i.e. by checking for the signature using the corresponding public key. Therefore the output of Γ' is distinguishable from the output of S'. This contradicts with our assumption that Γ' is a $\mathcal{P}$-code. Since Γ' cannot tell the output of S contains some sort of a digital signature or not, we conclude that Γ' must always output one of the c_i^*'s as its output.

We consider two cases. First if the entropy of $\mathcal{P}_h$ is at least $(1 + \epsilon)\log_2(t)$ for some fixed constant $\epsilon > 0$, then from the method of types (cf. [2, 4, 5]), we know that the c_i^*'s are distinct with overwhelming probability. Therefore Γ_1 achieves rate of $\log_2(t)$ bits per symbol. However, we know from previous paragraph that Γ' has its rate bounded by $\log_2(t)$. Hence in this case, Γ' does not do better than Γ_1.

In the second case, the entropy of $\mathcal{P}_h$ is at most $\log_2(t)$. In this case, method of types(cf. [2, 4, 5]) tell us that for any fixed constant $\delta > 0$ and large enought t, with overwhelming probability: the induced entropy of the view $(c_0^*, \ldots, c_{t-1}^*)$ is at least $(1-\delta)H(\mathcal{P}_h)$. Thus in this case our encoding Γ_1 achieves at least $(1-\delta)H(\mathcal{P}_h)$ bits per symbol. Note that the rate of the encoding Γ' must be bounded from above by $(1+\delta)H(\mathcal{P}_h)$, otherwise the output of Γ' will be distinguishable from $\mathcal{P}_h$ with overwhelming probability by simply estimating the entropies of the two distributions [4, 5].

We conclude that all cases, for all $\delta > 0$ our encoding Γ_1's rate is within $(1-\delta)$ fraction of the best possible rate minus some negligible factor, i.e. Γ_1 is essentially optimal.

6 Conclusions

We have shown in this article:

- Introduction and construction of $\mathcal{P}$-codes, and their applications.
- Efficient general construction of public key steganographic schemes secure against chosen hiddentext attacks using public key exchange assuming no special conditions.
- Efficient general construction of private key steganographic schemes secure against chosen hiddentext attacks assuming the existence of a pseudo-random generator.

Our constructions are essentially optimal in many cases, and they are general constructions, producing no errors in extraction. Nevertheless, our solutions do not come for free, i.e. they require polynomially sampleable cover distributions. Readers are refered to [7, 8] for more discussions on this issue.

Acknowledgement

The author is thankful to useful discussions with Professor Kaoru Kurosawa which have helped improving the presentation of this paper.

References

1. Ross J. Anderson and Fabien A.P. Petitcolas. On the limits of steganography. *IEEE Journal of Selected Areas in Communications*, 16(4):474–481, May 1998.
2. C. Cachin. An information-theoretic model for steganography. In *Information Hiding, Second International Workshop, Proceedings (Lecture Notes in Computer Science 1525)*, pages 306–318. Springer-Verlag, 1998. Portland, Oregon, April 15–17.
3. Scott Craver. On public-key steganography in the presence of an active warden. In David Aucsmith, editor, *Information Hiding, Second International Workshop, Portland, Oregon, USA*, volume 1525 of *Lecture Notes in Computer Science*. Springer, April 14-17 1998.
4. Csiszar. The method of types. *IEEETIT: IEEE Transactions on Information Theory*, 44, 1998.
5. I. Csiszar and J. Korner. *Information theory: Coding Theory for Discrete Memoryless Systems*. Academic Press, NY, 1981.
6. G. Gurther. A universal algorithm for homophonic coding. In *Eurocrypt '88*. Springer-Verlag, 1988.
7. Nick Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In Moti Young, editor, *Advances in Cryptoglogy — Crypto 2002, Proceedings*, volume 2442 of *LNCS*. Springer-Verlag, August 2002.

8. Nick Hopper and Luis von Ahn. Public key steganography. Submitted to Crypto 2003, http://www.cs.cmu.edu/˜ hopper.
9. S. Katzenbeisser and F. Petitcolas. On defining security in steganographic systems, 2002.
10. Mittelholzer. An information-theoretic approach to steganography and watermarking. In A. Pfitzmann, editor, *Proceedings of Third International Workshop on Information Hiding*, volume 1768 of *LNCS*. Springer-Verlag, September 1998.
11. P. Moulin and J. O'Sullivan. Information-theoretic analysis of information hiding, 1999.
12. Leonid Reyzin and Scott Russell. More efficient provably secure steganography. Technical report, IACR ePrint Archive 2003/093, 2003.
13. G. J. Simmons. The prisoner's problem and the subliminal channel. In David Chaum, editor, *Advances in Cryptology: Proceedings of Crypto '83*, pages 51–70, New York, USA, 1984. Plenum Publishing.
14. Jan Zollner, Hannes Federrath, Herbert Klimant, Andreas Pfitzmann, Rudi Piotraschke, Andreas Westfeld, Guntram Wicke, and Gritta Wolf. Modeling the security of steganographic systems. In *Information Hiding*, pages 344–354, 1998.