# Provably Secure Public-Key Steganography Based on Admissible Encoding

Xin Zhang, Kejiang Chen, *Member, IEEE*, Na Zhao, Weiming Zhang, *Member, IEEE*, and Nenghai Yu, *Member, IEEE*

***Abstract*—The technique of hiding secret messages within seemingly harmless covertext to evade examination by censors with rigorous security proofs is known as provably secure steganography (PSS). PSS evolves from symmetric key steganography to public-key steganography, functioning without the requirement of a pre-shared key and enabling the extension to multi-party covert communication and identity verification mechanisms. Recently, a public-key steganography method based on elliptic curves was proposed, which uses point compression to eliminate the algebraic structure of curve points. However, this method has strict requirements on the curve parameters and is only available on half of the points. To overcome these limitations, this paper proposes a more general elliptic curve public key steganography method based on admissible encoding. By applying the tensor square function to the known well-distributed encoding, we construct admissible encoding, which can create the pseudo-random public-key encryption function. The theoretical analysis and experimental results show that the proposed provable secure public-key steganography method can be deployed on all types of curves and utilize all points on the curve.**

***Index Terms*—Public-key steganography, elliptic curve cryptography, admissible encoding, provable security.**

## I. Introduction

STEGANOGRAPHY [1], [2], [3] is a covert communication method by embedding confidential data within ordinary media such as text, images, audio, and video. It protects the confidentiality of the information and conceals the presence of communication. The essence of steganography involves the steganographer placing secret data into common media to produce stegotext, aiming for these stegotext to be indifferent from the original media. Conversely, the attacker's task [4], referred to as steganalysis [5], is to identify the subtle differences between the original media and the stegotext [6].

Previous digital steganography techniques, such as least significant bit (LSB) replacement [7], exploiting modification direction (EMD) [8] and minimum distortion steganography [9], [10], primarily focused on empirical security without theoretical validation. These methods often fail against deep learning-based steganalysis attacks [11], [12].

A natural question arises: can steganography achieve a level of security comparable to that of cryptography? To address this question, Cachin [13] proposed the perfect security of steganography within an information-theoretic model, measured by the KL divergence $D_{\mathrm{KL}}(P_{\mathrm{c}}\|P_{\mathrm{s}})$, which remains an ideal model unachievable in practice. Hopper et al. [14] introduced a provably secure steganography based on computational complexity theory, aiming to prove that attackers with real-world capabilities cannot computationally distinguish between covertext and stegotext.

In Hopper's theory, he envisioned the concept of a perfect sampler, which has the capability of arbitrary sampling from the covertext distribution. Although this concept was not attainable then, with the development of deep learning and generative models [15], [16], [17] it has now become achievable. Researchers discovered that deep generative models can serve as perfect samplers, using data generated by these models as covertext to conceal information, thus constructing provably secure steganography. Several efforts have been made to use generative models with provably secure steganography, including AC [18], [19], ADG [20], Meteor [21], MEC [22] and Discop [23]. These works focus on the specific construction of embedding under the symmetric setting.

Recent research has increasingly focused on public-key steganography for many reasons. For instance, it does not rely on the assumption of a pre-shared key and offers significant efficiency advantages in multi-party covert communications [1], [14]. Furthermore, they can be extended to include identity verification mechanisms, further ensuring the security of communications. This paradigm shift marks a profound transition in steganography theory—from focusing on the design of individual methods to developing comprehensive communication protocols.

Following the conceptual framework of public-key steganography introduced by von Ahn and Hopper [24], Zhang et al. [25] advanced this domain by proposing a novel public-key steganography method based on elliptic curves. Their method leverages point compression to eliminate the algebraic structure of curve points, rendering the ciphertext produced by elliptic curve public-key encryption indistinguishable from random bit strings. This method not only addresses

Received 31 August 2024; revised 14 December 2024 and 17 February 2025; accepted 3 March 2025. Date of publication 10 March 2025; date of current version 25 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472398, Grant U2436601, Grant U2336206, and Grant 62402469. The associate editor coordinating the review of this article and approving it for publication was Prof. Fernando Perez-Gonzalez. *(Corresponding authors: Kejiang Chen; Weiming Zhang.)*

The authors are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230026, China, and also with Anhui Province Key Laboratory of Digital Security, Hefei 230009, China (e-mail: chenkj@ustc.edu.cn; zhangwm@ustc.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3550076

the challenge of embedding curve points directly into steganographic covertext but also presents advantages over the integer finite field-based methods proposed by von Ahn and Hopper [24] in terms of computational efficiency and encoding ciphertext size.

However, Zhang et al.'s method exhibits two limitations. First, it is constrained to work only on curves with a particular set of parameters, limiting its applicability across the broader range of elliptic curve parameters. This specificity allows attackers to potentially restrict the use of such curves. Second, the method is capable of utilizing only about half of the available curve points, necessitating the exclusion of the remaining points in practical applications. This restriction limits the deployment of algorithms and protocols that require the complete set of points, such as pairing-based methods [26], [27] and deterministic cryptographic protocols like BLS [28]. Consequently, this narrows the scope of public-key steganography for broader applications.

### A. Our Method

Inspired by the concept of admissible encoding in elliptic curve hash schemes [29], [30], [31], we introduce a more general elliptic curve public-key steganography method based on admissible encoding. This encoding method possesses excellent properties, allowing not only for a surjection over the entire set of elliptic curve points but also enabling the derivation of a distribution indistinguishable from the uniform distribution over a finite field when its sampleable inverse function is provided. Despite its appealing attributes, finding admissible encoding that works across a broad spectrum of elliptic curves presents a significant challenge. However, by applying the tensor square function to the known well-distributed encoding, we construct admissible encoding whose properties are suitable for creating the pseudo-random public-key encryption function.

We have discovered that well-distributed encodings which can be strengthened to admissible encodings are widely present across all types of curves. Through the tensor square detailed in the main text, we confirm that our proposed method can be effectively applied to all commonly used curves. To illustrate this, we utilize Icart encoding [29] on curves where $p = 2 \ mod \ 3$, SWU encoding [32] on curves where $p = 3 \ mod \ 4$, and SW [31] encoding on BN-like curves. We instantiate these three types of well-distributed encodings, construct their efficient sampleable inverse functions, and construct the corresponding public-key steganography schemes based on admissible encoding. Both our theoretical and experimental results prove the effectiveness and security of this scheme.

Furthermore, in Appendix A, we list a plethora of elliptic curve well-distributed encoding methods that can be applied within our framework to construct public-key steganography. Through the tensor exponent function, our scheme can be extended to even work on hyperelliptic curves.

**Contributions.** The main contributions of this paper can be summarized as follows:

- **Universal Applicability to All Types of Curve.** The new provable secure public-key steganography method we propose is deployable across all types of curves, significantly expanding the applicability of elliptic curve public-key steganography.
- **Full Utilization of Curve Points:** Our method utilizes all available points on the curve compared to the approach by Zhang et al. [25]. This comprehensive utilization facilitates the implementation of complex algorithms and protocols, including pairing operations and deterministic cryptographic protocols like the BLS signature scheme, all requiring access to the full set of curve points.
- **Efficient Instances of Commonly Used Curves.** In our instantiated schemes, we construct efficient sampleable inverse functions for the Icart, SW, and SWU methods. The corresponding public-key steganography instances operate on P-384, secp256k1, and P-256, respectively. Extensive statistical tests and steganalysis experiments validate the security of our constructions.

## II. Related Work

### A. Provably Secure Steganography

Provably secure steganography offers mathematically verifiable security to the steganography scheme, unlike its experience-based counterpart. It starts by defining a system model—symmetric or asymmetric, two-party or multi-party and then constructs a formal adversary model based on potential threats, mimicking real-world attacks. The approach uses rigorous math to reduce system security to some complex computational problems, ensuring the security is provable under assumptions.

Hopper et al. [14], [33] first introduced a framework of provably secure steganography by defining a probabilistic game named chosen plaintext attack (CPA), which models the scenario of a passive attack where the attacker hijacks the steganography encoder, which is also the working scenario for most steganalysis.

Following the definition, Hopper et al. proposed their construction, which is based on rejection sampling using a perfect sampler defined over the channel distribution and an unbiased function. Define a channel as a distribution with timestamp: $\mathcal{C} = ((c_1, t_1), (c_2, t_2), \ldots)$, the perfect sampler is an oracle $\mathcal{O}^{\mathcal{C}}$ providing exactly the distribution of $\mathcal{C}_h$, where $h$ is noted as history. A function $f : \mathcal{C} \rightarrow R$ is called $\epsilon - biased$ if $|Pr_{x \leftarrow \mathcal{C}}[f(x) = 0] - 1/|R|| \leq \epsilon$. $f$ is unbiased if $\epsilon = 0$. Given hiddentext $m$, the rejection sampling is defined as follows:

$$\text{sample} \quad x \quad \text{from} \quad \mathcal{O}^{\mathcal{C}} \quad \text{until} \quad f(x) = m.$$

Hopper et al. [14], [24] proved the security of their method against chosen plaintext attacks (CPA) by relying on the assumption of a perfect sampler and an unbiased function. However, the perfect sampler was not found at that time.

### B. Generative Model and Provably Secure Steganography

With the development of deep learning and generative models, researchers discovered that deep generative models can serve as perfect samplers, using data generated by these models as carriers to conceal information, thus constructing provably secure steganography.

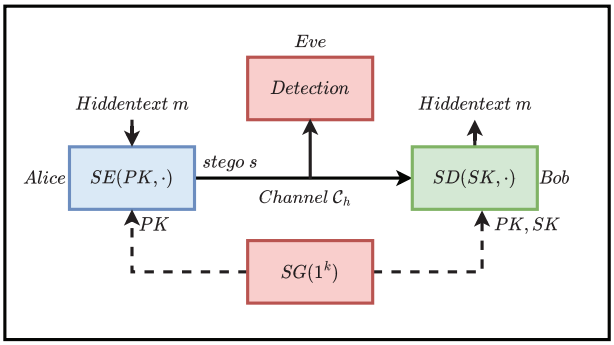



Fig. 1. Diagram of the public-key steganography system.

Numerous provably secure steganography methods have been devised. Yang et al. [34] pioneered provably secure steganography using autoregressive generative models and arithmetic coding (AC). Chen et al. [19] extended this to text-to-speech and text-generation tasks, respectively. Zhang et al. [20] introduced a method based on adaptive dynamic grouping (ADG) for provable security. Kaptchuk et al. [21] proposed Meteor to address randomness reuse in AC-based methods. Ding et al. [23] presented Discop, a more efficient method based on distribution copies. De Witt et al. [22] showed that maximal transmission efficiency in perfect security equals solving a minimum entropy coupling (MEC) problem. These works emphasize the specific construction of embedding in the symmetric, two-party setting and show potential extensions to the asymmetric, multi-party setting using public-key steganography.

### C. Provable Secure Public-Key Steganography

Recent research has shifted the focus from these symmetric steganography methods to asymmetric methods, namely public-key steganography. This approach operates without the need for a pre-shared key and allows expansion to multi-party covert communication and identity verification mechanisms.

*1) Definition:* As illustrated in Fig. 1, a public-key steganography system [24] comprises three probabilistic algorithms $SS = (SG, SE, SD)$. The algorithm $SG(1^k)$ generates a key pair $(PK, SK)$ from a random bitstream during the initial phase. The encoding function $SE(PK, m, h)$, using the public key $PK$, a hidden message $m$, and the history-based channel distribution $\mathcal{C}_h$, outputs stegotext $s_1, s_2, \ldots, s_l$ sampled from $\mathcal{C}_h$. The decoding function $SD$ takes the secret key $SK$, a sequence of stegotext $s_1, s_2, \ldots, s_l$, and the message history $h$, and returns the hidden message $m$. Both $SE$ and $SD$ have access to the channel $\mathcal{C}_h$.

*2) Existing Constructions:* Hopper et al. [35] refined the definition of the chosen plaintext attack (CPA, see Def. 9) for public-key steganography and pointed out that constructing public-key encryption functions that output pseudorandom ciphertexts is the core to creating public-key steganography. Building on this concept, they proposed pseudorandom public-key encryption methods based on RSA and Elgamal over integer finite fields, employing the probabilistic bias removal method (PBRM) to eliminate non-random probability biases. To address the computational and encoding inefficiencies of Hopper's method, Zhang et al. [25] proposed an elliptic curve public-key steganography based on point compression. Specifically, they constructed a bijection from approximately half of the curve points to a uniform random string on certain curves.

However, the previous point compression method et al. faces two major issues: it is only applicable to a small group of curves with specific parameters, namely, $E_{A,B} : By^2 = x^3 + Ax^2 + x \ (mod\ p, p \equiv 1\ (mod\ 4), \chi\left(A^2 - 4B\right) = -1, A\ B\left(A^2 - 4B\right) \neq 0.)$, which significantly narrows the scope of their method's applicability. Furthermore, it can only utilize about half of the curve points, forcing the exclusion of the remaining half of the points for practical deployment in public-key steganography. This limitation hinders the employment of algorithms and protocols that operate on the complete set of points, such as pairing [26], [27] and BLS protocol [28], thus constraining the utility of public-key steganography in a wider array of applications.

To address the two issues presented above, we found inspiration in the elliptic curve hash scheme and introduced the potent notion of admissible encoding. Specifically, we devised a method that involves constructing a random uniform mapping from a high-dimensional finite field to an elliptic curve domain, along with a deterministic inverse mapping. These mappings are utilized to establish the framework for public-key steganography.

### D. Elliptic Curve Hash and Admissible Encoding

The elliptic curve hash scheme is utilized in numerous cryptosystems that necessitate hashing an ID or something similar into an elliptic curve point. Such hash functions can substitute for any utilized within cryptosystems that rely on the random oracle model. Brier et al. [32] have established a sufficient condition for the construction of a hash function into an elliptic curve to be indistinguishable from a random oracle. This condition applies to hash functions of the following form:

$$\mathfrak{H}(m) = F(\mathfrak{h}(m)), \tag{1}$$

where $F : S \rightarrow E\left(\mathbb{F}_p\right)$ is a deterministic encoding, and $\mathfrak{h}$ is seen as a random oracle to $S$. Assuming that $\mathfrak{h}$ is a random oracle, the construction is indifferentiable whenever $F$ is an admissible encoding into $E\left(\mathbb{F}_p\right)$.

*1) Admissible Encoding:* Admissible encoding is a powerful concept integral to the construction of elliptic curve hash schemes (for a detailed definition, see Def. 5). It has excellent attributes including *computability, regularity, and samplability*, permitting a surjection across the entire set of elliptic curve points and facilitating the generation of a distribution that is indistinguishable from the uniform distribution over a higher-dimensional finite field, especially when its sampleable inverse function is provided.

We have discovered that by leveraging the properties of admissible encoding, we can effectively construct public-key encryption functions that output pseudorandom ciphertexts, thus addressing the core problem of public-key steganography. The following sections will detail this scheme and provide rigorous proof.

*2) Instantiation:* In the instantiation of our construction, a major technical difficulty lies in the fact that admissible encoding can hardly be constructed explicitly. To the best of our knowledge, only a special class of supersingular curves with specific parameters has an explicit expression for admissible encoding [26].

Drawing on the theory by Farashahi et al. [30], by applying the tensor square function to known well-distributed encoding, we construct our admissible encoding from a two-dimensional finite field to the set of elliptic curve points. We employed various well-distributed encodings to cover commonly used curve parameters. Specifically, we utilize Icart encoding [29] on curves where $p \equiv 2 \mod 3$, SWU encoding [32] on curves where $p \equiv 3 \mod 4$, and SW encoding [31] on BN-like curves. The corresponding public-key steganography instances operate on P-384, secp256k1, and P-256, respectively. In the Appendix, we also provide an extensive list of elliptic curve well-distributed encoding that can be integrated into our framework to construct public-key steganography. Through tensor square or tensor exponentiation, our scheme can be expanded to also hyperelliptic curves. Thus, we thoroughly demonstrate that our method can be applied to **all types of curves**, and due to the surjective nature of admissible encoding, our approach can utilize **all points on the curve**.

## III. Definition

*Definition 1 (Negligible Function):* A function $f : \mathbb{N} \to [0, 1]$ is negligible if for any polynomial $poly(\cdot)$, there exists a natural number $N \in \mathbb{N}$, $s.t.\ \forall\ n > N$, $f(n) < \frac{1}{poly(n)}$.

*Definition 2 (Statistical Indistinguishable):* Let $X$ and $Y$ be two random variables over a set $S$. The distributions of $X$ and $Y$ are $\epsilon$-statistically indistinguishable if:

$$\sum_{s \in S} |\Pr[\mathrm{X} = \mathrm{s}] - \Pr[\mathrm{Y} = \mathrm{s}]| \leq \varepsilon. \tag{2}$$

*Definition 3 (Basic Provably Secure Steganography Encoder:)* Let $\mathcal{C}_h$ denote the data distribution of the generative model given history $h$. Let $\mathcal{E}$ be a steganography encoder, with an output of maximum length $l$, and let $\mathcal{D}$ denote the corresponding decoding method. Assume $\mathcal{E}$ is $\epsilon$-statistically indistinguishable from the distribution of the channel $\mathcal{C}_h$, namely:

$$\begin{aligned} &\mathcal{E} : m \in \{0,1\}^t \to (s_1, \cdots, s_l) \in \mathcal{C}_h^l \\ &\sum_{c \in \mathcal{C}_h^l} |\Pr[\mathcal{C}_h^l = (s_1, \cdots, s_l)] - \Pr[\mathcal{C}_{\mathrm{h}}^{\mathrm{l}} = \mathrm{c}]| \leq \varepsilon \mathrm{l}. \end{aligned} \tag{3}$$

The probability is calculated over uniformly distributed $t$-bit strings and accounts for all randomness in $\mathcal{E}$. As shown in [14], there exist constructions of provably secure basic steganography encoders, such as through rejection sampling. Additionally, as mentioned in Section II-B, many private-key steganography methods can achieve negligible $\epsilon$-statistical indistinguishability.

*Definition 4 (Decisional Diffie-Hellman Assumption in Elliptic Curve Group:)* Let $G \triangleq E_{A,B}(\mathbb{F}_p)$ be a prime-order group of elliptic curve points, where $g$ is the generator and the order of the group is a prime $q$. Let $\mathcal{A}$ be a probabilistic polynomial-time machine (PPTM) that takes three elements from the group $G$ as input and outputs a single bit. The DDH advantage of $\mathcal{A}$ over the tuple $(G, g, q)$ is defined as:

$$\mathbf{Adv}^{\mathrm{ddh}}_{G,g,q}(\mathcal{A}) \triangleq \left| \begin{array}{c} \Pr\limits_{a,b}\left[\mathcal{A}\left(a \cdot g, b \cdot g, ab \cdot g\right) = 1\right] \\ -\Pr\limits_{a,b,c}\left[\mathcal{A}\left(a \cdot g, b \cdot g, c \cdot g\right) = 1\right] \end{array} \right|, \tag{4}$$

where $a, b, c$ are chosen uniformly at random from $\mathbb{Z}_q$.

The decisional Diffie-Hellman assumption in the Elliptic curve group is a computational hardness assumption requiring that $\mathbf{InSec}^{\mathrm{ddh}}_{G,g,q}(t) \triangleq \max_{\mathcal{A} \in \mathcal{A}(t)}\{\mathbf{Adv}^{\mathrm{ddh}}_{G,g,q}(\mathcal{A})\}$ is negligible in $k$.

*Definition 5 (Admissible Encoding):* [32] A function $F : S \to R$ between finite sets is an $\epsilon$-admissible encoding if it satisfies the following properties:

- **Computability**: $F$ is computable in deterministic polynomial time.
- **Regularity**: For $s$ uniformly distributed in $S$, the distribution of $F(s)$ is $\epsilon$-statistically indistinguishable from the uniform distribution in $R$.
- **Samplability**: exists an efficient randomized algorithm $\mathcal{I}$ such that for any $r \in R$, $\mathcal{I}(r)$ induces a distribution that is $\epsilon$-statistically indistinguishable from the uniform distribution in $F^{-1}(r)$.

$F$ is an admissible encoding if $\epsilon$ is a negligible function of the security parameter.

According to the definition, namely, the regularity gives:

$$\sum_{r \in R} \left| \Pr_s[F(s) = r] - \frac{1}{\#R} \right| = \sum_{r \in R} \left| \frac{\#F^{-1}(r)}{\#S} - \frac{1}{\#R} \right| \leq \varepsilon \tag{5}$$

where $\#R$, $\#S$, and $\#F^{-1}(r)$ correspond to the cardinalities of the sets $R$, $S$, and $F^{-1}(r)$, respectively.

Intuitively, an admissible encoding is a uniform and invertible mapping from the set of preimages to the set of images. Uniform sampling over the set of preimages, when passed through the admissible encoding, results in uniform sampling over the set of images. Given an image, all corresponding preimages can be determined.

*Definition 6 (Well-distributed Encoding):* [30] A function $f : \mathbb{F}_p \to E\left(\mathbb{F}_p\right)$ is said to be $B$-well-distributed for some $B > 0$ if, for all nontrivial characters $\chi$ of $E(\mathbb{F}_p)$, the following bound holds:

$$\left|S_f(\chi)\right| \leq B\sqrt{p}, \quad \text{where } S_f(\chi) = \sum_{u \in \mathbb{F}_p} \chi(f(u)). \tag{6}$$

*Definition 7 (Tensor Exponent and Tensor Square):* Given the curve $E\left(\mathbb{F}_p\right)$ over the finite field $\mathbb{F}_p$, the tensor exponent function $f^{\otimes s}$ is defined as follow:

$$\begin{aligned} f^{\otimes s} &: \mathbb{F}_p^s \to E\left(\mathbb{F}_p\right) \\ (u_1, \cdots, u_s) &\mapsto f(u_1) + \cdots + f(u_s). \end{aligned} \tag{7}$$

Consider for a given element $D \in E\left(\mathbb{F}_p\right)$, the number of preimage of $f^{\otimes s}$ is defined as:

$$\begin{aligned} N_s(D) \triangleq \#\big\{&(u_1, \cdots, u_s) \in \mathbb{F}_p^s \mid \\ &D \triangleq f(u_1) + \cdots + f(u_s)\big\}. \end{aligned} \tag{8}$$

Set $s$ to 2; the function is then referred to as the tensor square function:

$$f^{\otimes 2} : \mathbb{F}_p^2 \to E\left(\mathbb{F}_p\right)$$
$$(u, v) \mapsto f(u) + f(v). \tag{9}$$
$$N_2(D) \triangleq \#\left\{(u, v) \in \mathbb{F}_p^2 \mid D \triangleq f(u) + f(v)\right\}. \tag{10}$$

*Definition 8 (Pseudorandom Public-Key Encryption):* Following Hopper's theory [14], we construct pseudorandom public-key encryption schemes that are secure in a slightly nonstandard model, denoted by IND$-CPA, as opposed to the more standard IND-CPA. The key difference is that IND$-CPA requires the ciphertext output by the encryption algorithm to be indistinguishable from uniformly chosen random bits, while IND-CPA only requires that an adversary cannot distinguish the encryption of two chosen plaintexts. Importantly, IND$-CPA implies IND-CPA, but the converse does not hold, making IND$-CPA a strictly stronger requirement. This higher standard of security is particularly suitable for applications like steganography, where indistinguishability from random noise is critical.

Consider a public-key cryptography system $CS = (G, E, D)$ and a chosen plaintext attacker $\mathcal{A}$. $\mathcal{A}$ is allowed to play a game described as follows:

- **Key generation stage.** $(PK, SK) \leftarrow G(1^k)$.
- **Learning stage.** $\mathcal{A}$ sends plaintext $m_{\mathcal{A}}$ to the oracle and returns $E(PK, m_{\mathcal{A}})$. $\mathcal{A}$ can perform this stage multiple times.
- **Challenge stage.** $\mathcal{A}$ sends hiddentext $m \in \mathcal{M} \setminus \{m_{\mathcal{A}}\}$ to the oracle, which will flip a coin $b \in \{0, 1\}$. If $b = 0$, $\mathcal{A}$ obtains $c = E(PK, m)$; If $b = 1$, $\mathcal{A}$ obtains $u \leftarrow U_{|E(PK,\cdot)|}$.
- **Guessing stage.** $\mathcal{A}$ output a bit $b'$ as a "guess" about whether it obtains a plaintext or a random string.

Define the Chosen Plaintext Attack (CPA) advantage of $\mathcal{A}$ against $S$ by:

$$\mathbf{Adv}_{CS}^{\mathrm{cpa}}(\mathcal{A}, k) \triangleq \left| \begin{array}{c} \Pr\limits_{PK}\left[\mathcal{A}(PK, c) = 1\right] \\ -\Pr\limits_{PK}\left[\mathcal{A}(PK, u) = 1\right] \end{array} \right|. \tag{11}$$

A public-key encryption system is indistinguishable from uniformly random bits under chosen plaintext attack (IND$-CPA) if $\mathbf{InSec}_{CS}^{\mathrm{cpa}}(t, l, k) \triangleq \max_{\mathcal{A} \in \mathcal{A}_{(t,l)}}\{\mathbf{Adv}_{CS}^{\mathrm{cpa}}(\mathcal{A}, k)\}$ is negligible in $k$.

*Definition 9 (Chosen Hiddentext Attack):* Refer to Hopper et al. [35] and Zhang et al.'s paper [25], the **Threat Model** of public-key steganography is defined as follows:

Consider a public-key steganography system $SS = (SG, SE, SD)$ and an attacker $\mathcal{A}$. $\mathcal{A}$ play a game named chosen hiddentext attack (CHA) described as follows:

- **Key generation stage.** $(PK, SK) \leftarrow SG\left(1^k\right)$.
- **Learning stage.** $\mathcal{A}$ sends hiddentext $m_{\mathcal{A}}$ and history $h_{\mathcal{A}}$ and gets return $SE(PK, m_{\mathcal{A}}, h)$. $\mathcal{A}$ can perform this stage multiple times.
- **Challenge stage.** $\mathcal{A}$ sends hiddentext $m \in \mathcal{M} \setminus \{m_{\mathcal{A}}\}$ to an oracle, which will flip a coin $b \in \{0, 1\}$. If $b = 0$, $\mathcal{A}$ obtains $s = SE(PK, m, h)$; if $b = 1$, $\mathcal{A}$ obtains $c \leftarrow \mathcal{C}_h$.
- **Guessing stage.** $\mathcal{A}$ outputs a bit $b'$ as its "guess" to determine whether it has received a stegotext or a covertext.

Define the Chosen Hiddentext Attack (CHA) [14] advantage of $\mathcal{A}$ against $SS$ over channel $\mathcal{C}$ by:

$$\mathbf{Adv}_{SS,\mathcal{C}}^{\mathrm{cha}}(\mathcal{A}, k) \triangleq \left| \begin{array}{c} \Pr\limits_{PK}\left[\mathcal{A}(PK, s) = 1\right] \\ -\Pr\limits_{PK}\left[\mathcal{A}(PK, c) = 1\right] \end{array} \right|. \tag{12}$$

Define the *insecurity* of $SS$ over channel $\mathcal{C}$ by

$$\mathbf{InSec}_{SS,\mathcal{C}}^{\mathrm{cha}}(t, l, k) \triangleq \max_{\mathcal{A} \in \mathcal{A}_{(t,l)}}\{\mathbf{Adv}_{SS,\mathcal{C}}^{\mathrm{cha}}(\mathcal{A}, k)\}, \tag{13}$$

where $\mathcal{A}_{(t,l)}$ is the set of all adversaries that send at most $l(k)$ bits and run in time $t(k)$. $l(k)$ and $t(k)$ are polynomials of $k$. $SS$ is secure against CHA if $\mathbf{InSec}_{SS,\mathcal{C}}^{\mathrm{cha}}(t, l, k)$ is negligible in $k$, i.e., no probabilistic polynomial time (PPT) adversary can distinguish $s$ and $c$ with nonnegligible probability.

In this passive attack, the adversary $\mathcal{A}$ gains control over a steganographic encoder in the learning phase. $\mathcal{A}$ can embed specific hiddentext $m_{\mathcal{A}}$ into various stegotext. During the challenge phase, $\mathcal{A}$ receives a sample that might be either a stegotext generated with the encoder or a random cover. The objective in the guessing phase is for $\mathcal{A}$ to distinguish between covertext and stegotext with better accuracy than random chance. This model highlights the risks in steganography when the encoder of steganography is compromised, covering a wide range of steganalysis threats.

## IV. Our Proposed Method

We will expound on our provably secure public-key steganography scheme based on admissible encoding through three sequential steps IV-A, IV-B, and IV-C.

Initially, we will introduce our public-key steganography framework based on admissible encoding in IV-A. The intuition is as follows: To construct a mapping from elliptic curve points to pseudorandom bitstrings, the typical approach is to first map the curve points to a finite field $\mathbb{F}_p$ and then add redundancy. If our curve parameters do not allow for mapping the curve points to a single finite field, we consider constructing an admissible encoding to map the curve points to a two-dimensional or higher-dimensional finite field. In this section, we will detail how to construct admissible encoding from well-distributed encoding through tensor square, and then use admissible encoding to construct provably secure public-key steganography. Subsequently, we will present comprehensive proofs related to the aforementioned construction in IV-B. We will not only prove the effectiveness of the admissible encoding construction but also demonstrate that our final framework can withstand CHA attacks refer to Def. 9 in the random oracle model. Finally, we will explain how we instantiated our framework in IV-C. We adopted three different known well-distributed encodings including Icart [29], SW [31], [36] and SWU [37], constructed efficient sampleable inverse functions for these three encodings, and deployed our public-key steganography system on curves of three types of parameters. Subsequent experiments (V) on these three instances fully demonstrated the security and effectiveness of our entire system.

### A. Provably Secure Public-Key Steganography Framework Based on Admissible Encoding

*1) Well-Distributed Encoding Strengthen to Admissible Encoding:* Since the admissible encoding can hardly be constructed explicitly. Only a special class of supersingular curves with specific parameters has an explicit expression for admissible encoding. To construct public-key steganography that is general to all types of curves, we consider the possibility of constructing admissible encoding from a two-dimensional finite field to the finite field of elliptical curves. We utilize the tensor square function to strengthen well-distributed encoding on curves with a genus of 1 into admissible encoding (for constructions on curves not of genus 1, see Appendix).

Given a computable $B$-well-distributed encoding function $f : \mathbb{F}_p \to E(\mathbb{F}_p)$, the admissible encoding $F$ from the two-dimensional finite field $\mathbb{F}_p^2$ to the finite field of elliptic curves $E(\mathbb{F}_p)$ of genus 1 using the tensor square function as follows:

$$\begin{aligned} &F : \mathbb{F}_p^2 \to E\left(\mathbb{F}_p\right) \\ &F(u, v) = f^{\otimes 2} = f(u) + f(v). \end{aligned} \tag{14}$$

The complete proof of the admissibility of $F(u, v)$ can be found in Section IV-B, Lemma 1.

*2) Provably Secure Public-Key Steganography Framework Based on Admissible Encoding:* As illustrated in II-C, we will present a public-key steganography system consisting of three probabilistic algorithms, denoted as $SS = (SG, SE, SD)$.

Let $k$ be the security parameter. Let $E(\mathbb{F}_p)$ be the group of points on the elliptic curve of genus 1 defined over the finite field $\mathbb{F}_p$, where $p$ is a $k$-bit prime number. Let $g$ be a generator of the group of points on the curve, with order $q$, where $q$ is an $n$-bit prime number. Given $f$ a computable $B$-well-distributed encoding function $f : \mathbb{F}_p \to E(\mathbb{F}_p)$ and suppose $\mathcal{I}$ is its sampleable inverse function, namely

$$\begin{aligned} &\mathcal{I} : E(\mathbb{F}_p) \to \mathbb{F}_p \\ &\qquad P \to u \in D(P) = \left\{u \in \mathbb{F}_p \mid P = f(u)\right\}. \end{aligned} \tag{15}$$

**Algorithm 1** Public-key Steganography Key Pair Generation (SG)

**INPUT:** $1^k \in U(|k|), (p,\ E\left(\mathbb{F}_p\right),\ g,\ q)$
**OUTPUT:** $PK,\ SK$
1: Pick $x \in [0,\ q-1]$ at random;
2: $PK = x \cdot g,\ SK = x$

Suppose $k$ is the security parameter. The public-key steganography key pair generation (SG) is defined as Alg. 1:

Define $(E_K, D_K)$ as encryption and decryption functions of a private-key encryption scheme satisfying IND$-CPA, keyed by $\kappa$-bits key ($\kappa \le k$). Let $H$ be a cryptographically secure hash function $H : \{0,1\}^k \to \{0,1\}^\kappa$. In theoretical analysis, we model $H$ as a random oracle, an idealized function that returns an independently and uniformly distributed value for each unique input. In practice, $H$ will be instantiated with SHA-256 or another fixed cryptographic hash function. As defined in Def. 3, let $\mathcal{E}$ be a basic provably secure steganography encoder that achieves negligible $\epsilon$-statistical indistinguishability. The public-key steganography encoder (SE) is defined as Alg. 2.

**Algorithm 2** Public-key Steganography Encoder (SE)

**INPUT:** $\mathcal{E},\ m,\ (p,\ E\left(\mathbb{F}_p\right),\ g,\ q,\ PK),\ (f,\ \mathcal{I})$
**OUTPUT:** $s_1, s_2, \cdots, s_*$
1: ## *Key Deriving*
2: Pick $a \in [0,\ q-1]$ at random;
3: $P = a \cdot g$
4: $K = H(a \cdot PK)$
5: ## *Point Hiding*
6: **while** True **do**
7: Pick $v \in \mathbb{F}_p$ at random;
8: $D \leftarrow \mathcal{I}(P - f(v))$;
9: pick $i$ uniformly at random in $\#D$;
10: $u \leftarrow$ $i$-th element of $D$;
11: **if** $u = \varnothing$ **continue**;
12: **else break**;
13: **end while**
14: ## *Bias Eliminating*
15: Choose $t$-bit redundancy:
16: Pick $r_1 \in \left\{0, \ldots, \left\lfloor \frac{2^{k+t}-u}{p} \right\rfloor\right\}$
17: Pick $r_2 \in \left\{0, \ldots, \left\lfloor \frac{2^{k+t}-v}{p} \right\rfloor\right\}$
18: $(\tilde{u},\ \tilde{v}) = (u + r_1 p,\ v + r_2 p)$
19: ## *Final Encoding*
20: $C_1 = (\tilde{u},\ \tilde{v})$
21: $C_2 = E_K(m)$
22: $s_1,\ s_2, \cdots,\ s_* = \mathcal{E}(C_1 \| C_2)$

The steganography encoder (SE) consists of four parts. The first part is the temporary key deriving, which samples a group element randomly using a generator of the group and multiplies its order by the receiver's public key to obtain a temporary key through the key-derived function. The second part is point hiding, which involves inverse random sampling from curve points to the two-dimensional finite field using a sampleable inverse function $f$ of well-distributed encoding. During the Point Hiding step, a random index is used to select one point from these results. If the selected point is 'None', the algorithm resamples a new field element $v$, computes $f(v)$, and then calculates the inverse of the difference $P - f(v)$ using the sampleable inverse function again. This ensures that a valid point is eventually chosen through random sampling. Due to the admissible encoding properties, this part's expected running time is $O(1)$ field operations, ultimately yielding two randomly uniform finite field elements, $u$, and $v$. The third part is Bias Elimination, which expands the finite field to $k + t$ bits through redundant mapping to reduce the distribution distance after truncation to binary. The parameter $t$ in the Bias Eliminating step is set to $\frac{k}{4}$ or $\frac{k}{8}$ to strike a balance between security and efficiency. Larger $t$ (e.g., $t = \frac{k}{4}$) leads to a smaller statistical distance between the encoded field elements and a uniform random bit string, However, increasing $t$ also adds more bits to the ciphertext, which in turn increases both storage and transmission requirements. The fourth part is the final embedding process of steganography.

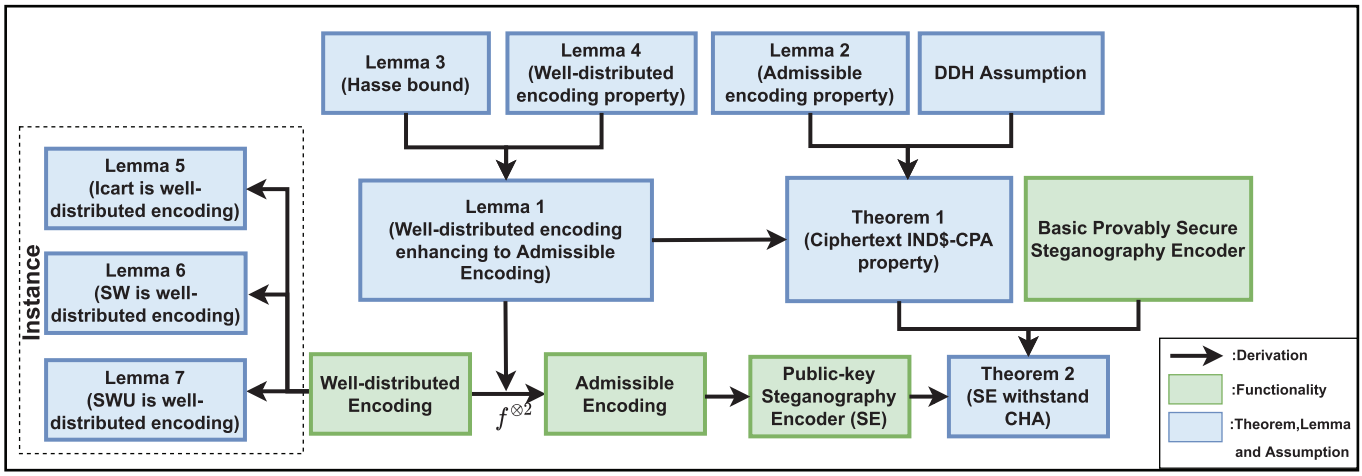



Fig. 2. The theoretical framework of our public-key steganography system.

**Algorithm 3** Public-key Steganography Decoder (SD)

**INPUT:** $\mathcal{D},\ s_{1..*},\ (p,\ E(\mathbb{F}_p),\ g,\ n,\ SK),\ f$
**OUTPUT:** $m$

$C = \mathcal{D}(s_{1..*})$ into $C_1,\ C_2$;
$(\tilde{u},\ \tilde{v}) = C_1$
$(u,\ v) = (\tilde{u}\ mod\ p,\ \tilde{v}\ mod\ p)$
$P = f(u) + f(v)$
$K = H(SK \cdot P)$
$m = D_K(K, C_2)$

The public-key steganography decoder (SD) is defined as Alg. 3.

The steganography decoder (SD) is straightforward. It splits the bit string extracted from the steganography, filters out the redundant parts using the module $q$, and decrypts the ciphertext by mapping the resulting finite field elements to curve points through admissible encoding.

### B. Proof of Security

In this subsection, we will provide detailed proof of the security of our constructed public-key steganography system SS = (SG, SE, SD) against CHA attacks as referred to in Def. 9. As shown in Fig. 2, We will present the proofs of two lemmas and two theorems. Lemma 1 explains the feasibility of constructing admissible encodings from well-distributed encoding via the tensor square method. Lemma 2 explains the excellent property of admissible encoding having uniform inverse sampling over the function's pre-image space. Theorem 1, combined with these two lemmas and DDH assumption, establishes that the ciphertext steganography encoder used in our system has the IND$-CPA property in the random oracle model. Finally, we present the conclusive proof of our system's security against CHA attacks in Theorem 2.

*Lemma 1:* Consider the tensor square function $f^{\otimes 2}$ defined as Def. 7, if $f$ is a $B$-well-distributed encoding, and is both computable and $\epsilon'$-sampleable, where $B$ is a constant and $\epsilon'$ is negligible relative to the security parameter, then $f^{\otimes 2}$ is an admissible encoding from $\mathbb{F}_p^2$ to $E(\mathbb{F}_p)$.

*Proof:* Refer to Def. 5, We will prove the three criteria of $f^{\otimes 2}$, namely *computability, regularity, and samplability*. The criterion of computability is trivial for the computability of $f$.

Consider the criterion of regularity, the number of preimage of $f^{\otimes 2}$ is $N_2(D) = \#\left\{(u,v) \in \left(\mathbb{F}_p\right)^2 \mid D = f(u) + f(v)\right\}$. Since $f$ is $B$-well-distributed encoding, according to Lemma 3 and Lemma 4, the statistical distance between the distribution of $F(u,v)$ for uniform $(u,v)$ and the uniform distribution on the curve can be bounded as:

$$\sum_{D \in E(\mathbb{F}_p)} \left| \frac{N_2(D)}{p^2} - \frac{1}{\#E(\mathbb{F}_p)} \right| \le \frac{B^2}{p} \sqrt{\#E(\mathbb{F}_p)} \le \frac{B^2}{p}(\sqrt{p}+1) \le 2B^2 p^{-\frac{1}{2}}, \tag{16}$$

which is a negligible function as $B$ is constant. This proves $\epsilon$-regularity.

Consider the criterion of samplability, since $f$ is $\epsilon'$-sampleable, we denote $\mathcal{I}$ as its sampleable inverse function:

$$\mathcal{I} : E(\mathbb{F}_p) \to \mathbb{F}_p$$
$$P \to u \in D(P) = \left\{u \in \mathbb{F}_p \mid P = f(u)\right\}. \tag{17}$$

To show the samplability of $f^{\otimes 2}$, we construct the sampling algorithm for $f^{\otimes 2}$ as Alg. 4.

**Algorithm 4** Sampling algorithm for $f^{\otimes 2}$

**INPUT:** $P \in E(\mathbb{F}_p),\ \mathcal{I}$
**OUTPUT:** $(u,v) \in D^2(P) = \left\{(u,v) \in (\mathbb{F}_p)^2 \mid P = f^{\otimes 2}(u,v)\right\}$

**while** True **do**
  Pick $v \in \mathbb{F}_p$ at random;
  $D \leftarrow \mathcal{I}(P - f(v))$;
  pick $i$ uniformly at random in $\#D$;
  $u \leftarrow$ $i$-th element of $D$;
  **if** $u = \varnothing$ **continue**;
  **else return** $(u,v)$;
**end while**

For well-distributed encoding, the number of unmapped points is bounded. Consequently, the number of repetitions

is polynomially bounded. Given that $\epsilon'$ is negligible, the computable algorithm ensures a uniform distribution of $D^2(P)$, thereby demonstrating $\epsilon$-regularity. □

*Lemma 2:* Given an $\epsilon$-admissible encoding $F : S \to R$ between finite sets and its sampleable inverse function $\mathcal{I}$, for $r$ uniformly distributed in $R$, the reversed distribution of $s = \mathcal{I}(r)$ is $2\epsilon$-statistically indistinguishable from the uniform distribution in $S$.

*Proof:* Our target is to prove that for all randomness in $\mathcal{I}$, we have statistical distance between reversed distribution and uniform distribution is bounded as follows:

$$\begin{aligned}\delta &:= \sum_{s\in S}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{1}{\#S}\right| \\ &\le 2\epsilon\sum_{r\in R}\sum_{s\in F^{-1}(r)}\frac{1}{\#R}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{\#R}{\#S}\right| \\ &\le 2\epsilon\sum_{r\in R}\sum_{s\in F^{-1}(r)}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{\#R}{\#S}\right| \\ &\le \underbrace{\sum_{r\in R}\sum_{s\in F^{-1}(r)}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{1}{\#F^{-1}(r)}\right|}_{\delta_1} \\ &\quad+\underbrace{\sum_{r\in R}\sum_{s\in F^{-1}(r)}\left|\frac{1}{\#F^{-1}(r)}-\frac{\#R}{\#S}\right|}_{\delta_2}\le 2\epsilon, \end{aligned} \tag{18}$$

Since we have $\delta_2$

$$\begin{aligned}&=\sum_{r\in R}\sum_{s\in F^{-1}(r)}\left|\frac{1}{\#F^{-1}(r)}-\frac{\#R}{\#S}\right| \\ &=\sum_{r\in R}\left|\frac{\#F^{-1}(r)}{\#S}-\frac{1}{\#R}\right|=\sum_{r\in R}\left|\Pr_s[f(s)=r]-\frac{1}{\#R}\right|, \end{aligned} \tag{19}$$

which is the statistical distance between $F(s)$ and uniform distribution in $R$. According to the regularity of admissible encoding defined by Def. 5, we have $\delta_2 \le \epsilon$.

Regarding $\delta_1$, according to the samplability of admissible encoding, we have:

$$\begin{aligned}\sum_{s\in F^{-1}(r)}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{1}{\#F^{-1}(r)}\right|&\le\epsilon \\ \delta_1=\sum_{r\in R}\sum_{s\in F^{-1}(r)}\frac{1}{\#R}\left|\Pr_r[\mathcal{I}(r)=s]-\frac{1}{\#F^{-1}(r)}\right|&\le\epsilon \end{aligned} \tag{20}$$

Hence, $\delta = \delta_1 + \delta_2 \le 2\epsilon$. □

*Theorem 1:* Let $\mathcal{E}$ be a distribution that is $\epsilon$-statistically indistinguishable from the distribution of the channel $\mathcal{C}_h$. Let $f$ be a $B$-well-distributed encoding that is both computable and $\epsilon'$-sampleable, and $\mathcal{I}$ be a sampleable inverse function of $f$. Let $H$ be a cryptographically secure hash function $H : \{0,1\}^k \to \{0,1\}^\kappa$ which can be modeled as a random oracle. Under the decisional Diffie-Hellman (DDH) assumption in the elliptic curve group, the ciphertext $C_1\|C_2$ produced by Algorithm 2 is indistinguishable from uniformly random bits under a chosen plaintext attack (IND$-CPA security).

*Proof:* Define $H_0 \triangleq C_1\|C_2 = (\tilde{u},\ \tilde{v})\|E_{H(ax\cdot g)}(m)$, where $f(u)+f(v) = a\cdot g$.

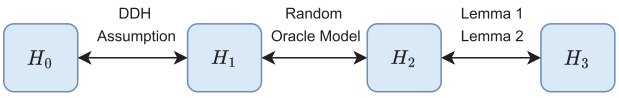



Fig. 3. The hardness of distinguishing between $H_0$ and $H_3$.

Define $H_1$ as the variant of $H_0$ where $ax\cdot g$ is replaced by a random element of the group $E(\mathbb{F}_p)$, i.e. $H_1 \triangleq C_1\|C_2' = (\tilde{u},\ \tilde{v})\|E_{H(c\cdot g)}(m)$.

Define $H_2$ as the variant of $H_1$ where $H(c\cdot g)$ is replaced by a random draw from $\{0,1\}^\kappa$, i.e. $H_2 \triangleq C_1\|C_2'' = (\tilde{u},\ \tilde{v})\|E_{r\in\{0,1\}^\kappa}(m)$.

Define $H_3$ as the variant of $H_2$ where $C_1$ is replaced by a random draw from $\{0,1\}^{2(k+t)}$, i.e. $H_3 \triangleq \tilde{r}\|C_2'' = \tilde{r}\|E_{r\in\{0,1\}^\kappa}(m)$, where $\tilde{r}\in\{0,1\}^{2(k+t)}$.

As shown in Fig. 3, We claim that the advantage of distinguishing between $H_0$ and $H_1$, $H_1$ and $H_2$, $H_2$ and $H_3$, as well as $H_3$ and random bits, are all negligible in $k$.

(1) Distinguishing $H_3$ from random bits requires distinguishing $E_K(m)$ from random bits, which contradicts the IND$-CPA security of the encryption scheme $E_K$.

(2) Distinguishing $H_2$ from $H_3$ would contradict Lemma 1 and Lemma 2, the reason is as follows:
Consider that $C_1 = (\tilde{u},\ \tilde{v}) = (u+r_1p,\ v+r_2p)$ represents the pre-image under the admissible encoding $F(u,v)$, where $F(u,v) = f^{\otimes 2} = f(u)+f(v)$. This construction leverages a B-well-distributed encoding $f$, applied through the tensor square. Accordingly, the admissibility of $F(u,v)$ is established as per Lemma 1, rendering $F(u,v)$ an $\tilde{\epsilon}$-admissible encoding. The value of $\tilde{\epsilon}$ is determined as $\tilde{\epsilon} = \max\left\{2B^2p^{-\frac{1}{2}}, \epsilon'\right\}$, thereby affirming the admissibility criteria.
According to Lemma 2, given a sampleable inverse function of a $\tilde{\epsilon}$-admissible encoding $F$, the distribution of the pre-image is $2\tilde{\epsilon}$-statistically indistinguishable from the uniform distribution over $\mathbb{F}_p^2$. The bias-eliminating process, as detailed in Algorithm 2, ensures that the statistical distance between the field $\mathbb{F}_p$ and a $k$-bit uniform random string is bounded by $2^{-t}$, evidenced by the equation:

$$\sum_{u\in\mathbb{F}_p}\left|\frac{\left\lfloor\frac{2^{k+t}-u}{p}\right\rfloor+1}{2^{k+t}}-\frac{1}{p}\right|\le\frac{p}{2^{k+t}}\le 2^{-t}, \tag{21}$$

From this, the statistical distance between $C_1$ and a $2(k+t)$-bit uniform random string is limited to $2\cdot2^{-t}+2\tilde{\epsilon}$. Moreover, the statistical distance between $C_2$ and $l$-bit uniform random string is $\epsilon_0$, assuming the block cipher $E_K$ is semantically secure. Consequently, the overall statistical distance is constrained by $\epsilon_c = 2\cdot2^{-t}+2\tilde{\epsilon}+\epsilon_0$. For practical applications, selecting $t$ as either $\frac{k}{4}$ or $\frac{k}{8}$ ensures that $\epsilon_c$ is negligible in $k$.

(3) The advantage of distinguishing $H_1$ from $H_2$ is negligible if $H$ is modeled as a random oracle $\mathcal{O}$, where the output is independently and uniformly distributed for each unique input. The reason is as follows:
In $H_1$, the input $c\cdot g$ is assumed to be a uniformly distributed random element (with $c$ being randomly

generated and $g$ being a fixed generator). Consequently, $\mathcal{O}(c \cdot g)$ is also uniformly distributed over $\{0,1\}^\kappa$.
In $H_2$, $H(c \cdot g)$ is directly replaced by a uniformly random value $r \in \{0,1\}^\kappa$.
Since the distributions of $H(c \cdot g)$ and $r$ are identical, then $H_1$ and $H_2$ are indistinguishable in the Random Oracle Model for any polynomial-time adversary.

(4) Distinguishing $H_0$ from $H_1$ would contradict the Decisional Diffie-Hellman (DDH) assumption in the elliptic curve group as defined in Def. 4. Suppose there exists a probabilistic polynomial time algorithm $\mathcal{A}$ that can distinguish between $H_0$ and $H_1$ with non-negligible probability $\epsilon$. In that case, we can construct another probabilistic polynomial time algorithm $\mathcal{A}'$ to break the DDH assumption.

The construction of $\mathcal{A}'$ is straightforward: when $\mathcal{A}'$ receives $(a \cdot g, b \cdot g, c \cdot g)$, it sets $PK = b \cdot g$, computes $C_1 \| C \triangleq (\tilde{u}, \tilde{v}) \| E_{H(c \cdot g)}(m)$, then runs $\mathcal{A}$ on $C_1 \| C$ and outputs its result. If $c = ab$, then $C_1 \| C = H_0$. If $c$ is chosen uniformly at random from the group, then $C_1 \| C = H_1$.

Thus, $\mathcal{A}'$ achieves at least $\epsilon/2$ advantage in distinguishing $(a \cdot g, b \cdot g, ab \cdot g)$ from $(a \cdot g, b \cdot g, c \cdot g)$. □

*Theorem 2:* Let $\mathcal{E}$ be a distribution that is $\epsilon$-statistically indistinguishable from the distribution of the channel $\mathcal{C}_h$. Let $f$ be a $B$-well-distributed encoding that is both computable and $\epsilon'$-sampleable, and $\mathcal{I}$ be a sampleable inverse function of $f$. Let $H$ be a cryptographically secure hash function $H : \{0,1\}^k \to \{0,1\}^\kappa$ which can be modeled as a random oracle. Under the decisional Diffie-Hellman (DDH) assumption in the elliptic curve group, the insecurity of the constructed public-key steganography system SS = (SG, SE, SD) against chosen hider attacks (CHA) is negligible.

*Proof:* Supposed there exists a probabilistic polynomial time algorithm $\mathcal{A}$ that can distinguish between stegotext $s$ and covertext $c$ with non-negligible probability, We can construct a probabilistic polynomial time algorithm $\mathcal{A}'$ which plays the IND$-CPA game: distinguishing $C_1 \| C_2$ from $U_{(2(k+t))}$.

The construction of $\mathcal{A}'$ is straightforward: $\mathcal{A}'$ first chooses history $h_\mathcal{A}$ and a message $m_\mathcal{A}$ and then runs $\mathcal{A}$ to go through the key generation stage. During the Challenge stage, $\mathcal{A}'$ picks plaintext $m \in \mathcal{M} \setminus \{m_\mathcal{A}\}$ and sends it to the oracle. The oracle will flip a coin $b$, where for $b = 0$, $\mathcal{A}'$ obtains $C_1 \| C_2$, and for $b = 1$, $\mathcal{A}'$ obtains $u \leftarrow U_{(2(k+t))}$. After receiving the oracle's return, $\mathcal{A}'$ encodes it into multimedia data using the generated model and sends it to $\mathcal{A}$ to make a guess about the coin flip. $\mathcal{A}$ outputs a bit $b'$ as its answer, which is also $\mathcal{A}'$'s answer. The total time of the whole process is $t + O(lk)$.

If $b = 0$, then $s \leftarrow \mathcal{E}(PK, m, h)$, so $\Pr[\mathcal{A}'(PK, C_1 \| C_2) = 1] = \Pr[\mathcal{A}(PK, s) = 1]$. If $b = 1$, then $c \leftarrow U_{(2(k+t))}$, so $s$ is distributed identically to $\mathcal{C}_h^l$. Thus, $|\Pr[\mathcal{A}'(PK, u) = 1] - \Pr[\mathcal{A}(PK, \mathcal{C}_h^l) = 1]| \le \epsilon l$ because $\mathcal{E}$ is $\epsilon$-statistically indistinguishable from the distribution of the channel $\mathcal{C}_h^l$.

Combining the cases, we have

$$\begin{aligned}
&|\Pr[\mathcal{A}(PK, s) - \Pr[\mathcal{A}(PK, \mathcal{C}_h^l) = 1]| \\
&\quad = |\Pr[\mathcal{A}'(PK, C_1 \| C_2) - \Pr[\mathcal{A}(PK, \mathcal{C}_h^l) = 1]| \\
&\quad \le |\Pr[\mathcal{A}'(PK, C_1 \| C_2) - \Pr[\mathcal{A}'(PK, u) = 1]| \\
&\qquad + |\Pr[\mathcal{A}'(PK, u) = 1] - \Pr[\mathcal{A}(PK, \mathcal{C}_h^l) = 1]| \\
&\quad \le \mathbf{Adv}_{CS}^{\mathrm{cpa}}(\mathcal{A}, k) + \epsilon l, \\
&\qquad i.e. \quad \mathbf{Adv}_{SS,C}^{\mathrm{cha}}(\mathcal{A}, k) \le \mathbf{Adv}_{CS}^{\mathrm{cpa}}(\mathcal{A}', k) + \epsilon l. \qquad (22)
\end{aligned}$$

Thus, if $\mathbf{Adv}_{SS,C}^{\mathrm{cha}}(\mathcal{A}, k)$ is non-negligible, then $\mathbf{Adv}_{CS}^{\mathrm{cpa}}(\mathcal{A}', k) \ge \mathbf{Adv}_{SS,C}^{\mathrm{cha}}(\mathcal{A}, k) - \epsilon l$ is also non-negligible, which contradicts IND$-CPA property proved in Theorem 1.

Hence, we have comprehensively completed the proof. □

### C. Instance

To demonstrate the generality of our framework and provide instances for the practical deployment of public-key steganography, we will explain how we instantiated our framework on commonly used curves. We adopted three different known well-distributed encodings including Icart [29], SW [31], [36] and SWU [37], constructed efficient sampleable inverse functions for these three encodings, and deployed our public-key steganography system on curves of three types of parameters, namely P-384, secp256k1 and P-256.

*1) Icart's Encoding:* Icart et al. [29] proposed an encoding method that utilizes the cube root of the curve equation, employing radicals whose degrees are prime relative to the order of the multiplicative group. Consider the curve $E_{a,b} : y^2 = x^3 + ax + b$ over the field $E_{a,b}(\mathbb{F}_p)$ where $p > 3$ and $p = 2 \pmod 3$. In these finite fields, the function $x \mapsto x^3$ is a bijection with inverse function $x \mapsto x^{1/3} = x^{(2p-1)/3}$. The Icart's encoding is defined as follows:

$$\begin{aligned}
f : \mathbb{F}_p &\mapsto E_{a,b}(\mathbb{F}_p) \\
u &\mapsto (x, y) \\
\text{where } x &= \left(v^2 - b - \frac{u^6}{27}\right)^{1/3} + \frac{u^2}{3}, \\
y &= ux + v, \\
v &= \frac{3a - u^4}{6u}. \qquad (23)
\end{aligned}$$

The Icart's encoding is a $(12 + 3p^{-\frac{1}{2}})$-well-distributed encoding according to lemma. 5. To construct its sampleable inverse function $\mathcal{I}$, it is necessary to solve the quartic equation over a finite field:

$$u^4 - 6u^2 x + 6uy - 3a = 0. \qquad (24)$$

We use the Berlekamp algorithm [38] to get the solution set, and $\mathcal{I}(P, i)$ return $i$-th root of the Eq. 24. Solving polynomial equations of degree $d$ over a finite field can be achieved in $O(d^3 + M(d) \cdot \log d \cdot \log p)$ scalar operations, where $M(d) = d \cdot \log d \cdot \log \log d$. In the case of a degree-4 polynomial ($d = 4$) and a $k$-bit prime $p$, this complexity becomes $O(k)$ scalar operations. Since scalar operations such as division, inversion, squareness check, and square root can be implemented in $O(k^3)$ within the finite field $\mathbb{F}_p$ assuming that multiplication is implemented in $O(k^2)$, the overall time complexity of the algorithm is $O(k^4)$.

For our entire public-key steganography system using Icart's encoding, we have chosen the P-384 elliptic curve. This curve is part of the NIST (National Institute of Standards and Technology) suite of standards for elliptic curve cryptography,

renowned for its strong security properties and efficiency in cryptographic operations. The choice of P-384 specifically offers a good balance between computational efficiency and security, making it well-suited for the demanding requirements of public-key steganography. The parameter specifications of the P-384 curve are as follows:

$$\text{P-384:} \begin{cases} p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1, \\ y^2 = x^3 - 3x + b. \end{cases} \tag{25}$$

*2) SW Encoding:* Shallue and van de Woestijne [36] construct an encoding function $f : \mathbb{F}_p \to E\left(\mathbb{F}_p\right)$ based on the construction of explicit rational curves on a surface associated with the target curve. It is worth emphasizing that this encoding can be used in any curves that can be expressed in the Weierstrass form: $E : y^2 = g(x) = x^3 + Ax^2 + Bx + C$. Here for simplicity, we consider SW encoding applied to a BN curve [39] with $E : y^2 = g(x) = x^3 + b$. The SW encoding is defined as follows:

$$\begin{aligned} f : \mathbb{F}_p &\mapsto E\left(\mathbb{F}_p\right) \\ t &\mapsto \left(x_i, \chi_p(t) \cdot \sqrt{g\left(x_i\right)}\right) \\ x_1(t) &= \frac{-1 + \sqrt{-3}}{2} - \frac{\sqrt{-3} \cdot t^2}{1 + b + t^2}, \\ x_2(t) &= \frac{-1 - \sqrt{-3}}{2} + \frac{\sqrt{-3} \cdot t^2}{1 + b + t^2}, \\ x_3(t) &= 1 - \frac{\left(1 + b + t^2\right)^2}{3t^2}. \end{aligned} \tag{26}$$

where for each $t$, $i \in \{1, 2, 3\}$ is the smallest index such that $g(x_i)$ is a square in $\mathbb{F}_p$.

**Algorithm 5** Sampleable Inverse Function $\mathcal{I}$ for SW Encoding

**INPUT:** $(x,\ y) = P \in E(\mathbb{F}_p) : y^2 = x^3 + b\ (mod\ p)$
**OUTPUT:** $D(P) = \left\{u \in \mathbb{F}_p \mid P = f(u)\right\}$

$c_1 = \sqrt{-3},\ c_2 = (c_1 - 1)/2,\ c_3 = (-c_1 - 1)/2$
$z = 2x + 1$
$s_1 = (1 + b)(c_1 - z)/(c_1 + z)$
$s_2 = (1 + b)(c_1 + z)/(c_1 - z)$
$s_3 = (z + \sqrt{(z^2 - 16(b + 1)^2)})/4$
$s_4 = (z - \sqrt{(z^2 - 16(b + 1)^2)})/4$
**If** $c_2 - \frac{c_1 s_1}{1+b+s_1}$ is square **then** $s_2 = s_3 = s_4 = \perp$;
**If** $c_3 + \frac{c_1 s_2}{1+b+s_2}$ is square **then** $s_3 = s_4 = \perp$;
$u_1, u_2, u_3, u_4 = \sqrt{s_1}, \sqrt{s_2}, \sqrt{s_3}, \sqrt{s_4}$
set $u_i = -u_i$ if $is\_odd(u_i) \neq is\_odd(y), \forall i \in 1..4$
**return:** $D(P) = \{u_1, u_2, u_3, u_4\}$

The SW encoding is a $(62 + O\left(p^{-\frac{1}{2}}\right))$-well-distributed encoding according to lemma. 6. To construct its sampleable inverse function $\mathcal{I}$, we need to solve for $t$ when given a curve point $(x, y)$. Accordingly, we present Alg. 5 as follows:

In Alg. 5, we search for a feasible solution for $t^2$ using the formulas for $x_1(t)$, $x_2(t)$, and $x_3(t)$ one by one. For each formula, we check whether $t^2$ is a square in $\mathbb{F}_q$. Since $i \in \{1, 2, 3\}$ is the smallest index such that $g(x_i)$ is a square in $\mathbb{F}_q$, we only take the first $t^2$ that satisfies this condition as our solution. Therefore, the correctness of the algorithm is established.

As for complexity, the algorithm we provided performs a series of calculations and checks to determine whether $t^2$ is a square in $\mathbb{F}_p$ for each of the three possible cases: $x_1(t)$, $x_2(t)$, and $x_3(t)$. Since each step (including addition, subtraction, multiplication, inversion, square root calculation, and squareness checking) requires a constant number of field operations, the algorithm's complexity is primarily determined by the most computationally expensive operations: square root calculation and squareness checking. The overall time complexity of the algorithm is $O(k^3)$.

For our entire public-key steganography system using SW encoding, we have chosen curves with characteristics similar to BN curves, including widely used ones like secp256k1, due to their potential suitability for pairing deployment. The parameter specifications of the P-384 curve are as follows:

$$\text{secp256k1:} \begin{cases} p = 2^{256} - 2^{32} - 997, \\ y^2 = x^3 + 7. \end{cases} \tag{27}$$

### D. SWU Encoding

Ulas [37] enhanced the SW encoding to diminish its complexity for curves defined by the equation $E : y^2 = g(x) = x^3 + ax + b$ where $a, b \neq 0$ and $p = 3(\mod 4)$. The SWU encoding is defined as follows:

$$\begin{aligned} f : \mathbb{F}_p &\mapsto E\left(\mathbb{F}_p\right) \\ t &\mapsto \left(x_i, \chi_p(t) \cdot \sqrt{g\left(x_i\right)}\right) \\ x_1(t) &= \frac{-b}{a}\left(1 + \frac{1}{t^4 - t^2}\right), \\ x_2(t) &= \frac{bt^2}{a}\left(1 + \frac{1}{t^4 - t^2}\right). \end{aligned} \tag{28}$$

where for each $t$, $i \in \{1, 2\}$ is the smallest index such that $g(x_i)$ is a square in $\mathbb{F}_p$.

**Algorithm 6** Sampleable Inverse Function $\mathcal{I}$ for SWU Encoding

**INPUT:** $(x,\ y) = P \in E(\mathbb{F}_p) : y^2 = x^3 + b\ (mod\ p)$
**OUTPUT:** $D(P) = \left\{u \in \mathbb{F}_p \mid P = f(u)\right\}$

$\delta_1 = 1 - \frac{4b}{ax+b}$
$\delta_2 = \left(\frac{ax}{b} + 1\right)^2 - 4\left(\frac{ax}{b} + 1\right)$
$s_1 = \frac{1-\sqrt{\delta_1}}{2},\ s_2 = \frac{1+\sqrt{\delta_1}}{2}$
$s_3 = \left(\frac{ax}{b} + 1 - \sqrt{\delta_1}\right)/2,\ s_4 = \left(\frac{ax}{b} + 1 + \sqrt{\delta_1}\right)/2$
**If** $\frac{-b}{a}\left(1 + \frac{1}{(s_1)^2 - s_1}\right)$ is square or $\frac{-b}{a}\left(1 + \frac{1}{(s_2)^2 - s_2}\right)$ is square **then** $s_3 = s_4 = \perp$;
$u_1, u_2, u_3, u_4 = \sqrt{s_1}, \sqrt{s_2}, \sqrt{s_3}, \sqrt{s_4}$
set $u_i = -u_i$ if $is\_odd(u_i) \neq is\_odd(y),\ \forall i \in 1..4$
**return:** $D(P) = \{u_1, u_2, u_3, u_4\}$

The SWU encoding is a $\left(52 + 151p^{-\frac{1}{2}}\right)$-well-distributed encoding according to lemma. 7. To construct its sampleable inverse function $\mathcal{I}$, we need to solve for $t$ when given a curve point $(x, y)$. Accordingly, we present Alg. 6 as follows:

In Alg. 6, we search for a feasible solution for $t^2$ using the formulas for $x_1(t)$ and $x_2(t)$. For each formula, we check whether $t^2$ is a square in $\mathbb{F}_p$. Since $i \in \{1, 2\}$ is the smallest

index such that $g(x_i)$ is a square in $\mathbb{F}_p$, we only take the first $t^2$ that satisfies this condition as our solution. Therefore, the correctness of the algorithm is established.

As for complexity, since each step (including addition, subtraction, multiplication, inversion, square root calculation, and squareness checking) requires a constant number of field operations, the algorithm's complexity is primarily determined by the most computationally expensive operations: square root calculation and squareness checking. The overall time complexity of the algorithm is $O(k^3)$.

For our entire public-key steganography system using SWU encoding, we have chosen the P-256 elliptic curve. This curve is part of the NIST suite of standardized curves. The parameter specifications of the P-256 curve are as follows:

$$\text{P-256: } \begin{cases} p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1, \\ y^2 = x^3 - 3x + b. \end{cases} \tag{29}$$

Comparing the algorithmic complexities of these three methods, SWU may offer better efficiency in terms of runtime due to its fewer field operations and deployment on smaller finite fields, and Icart generally requires the most field operations, which could lead to relatively the slowest performance. However, the key strength of these three methods lies in their applicability to different types of elliptic curves, ensuring that our approach can be extended to a wide range of curve parameters. This flexibility underscores the robustness and versatility of our methods in diverse elliptic curve settings.

Up to now, we have instantiated our public-key steganography framework on three categories of commonly used curves, employing three distinct methods of well-distributed encoding to develop admissible encodings, which has effectively demonstrated the versatility and efficiency of our proposed framework.

## V. Experiments

In this section, we evaluate the pseudorandomness of the proposed elliptic curve pseudorandom public-key encryption algorithm through statistical tests. Additionally, we validate the security of our proposed public-key steganography instance[1] through steganalysis experiments.

### A. Statistical Test for Pseudorandomness

To evaluate the pseudorandomness of our elliptic curve pseudorandom public-key encryption algorithm, we utilized the NIST SP 800-22 test suite.[2]

A key pair was generated by Alg. 1, and we generated the ciphertext $C_1 \| C_2$ by Alg. 2. This process is repeated to compile a binary string exceeding $10^8$ bits, then segmented into 100 equal-length streams for 15 statistical tests. This procedure was replicated with various key pairs, yielding consistent results. Below in Tab. I, we detail the outcomes from a representative trial:

[1]Our code can be found on https://github.com/XinZhang1999/Public-key-Discop

[2]You can download the NIST SP 800-22 test suite directly from https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software

TABLE I
Statistical Tests for Pseudorandomness of Ciphertext $C_1 \| C_2$ in Alg. 2 Using Icart's, SW, and SWU Methods

| Statistical Test | P-VALUE | | | Result |
|---|---|---|---|---|
| | Icart's | SW | SWU | Combined |
| Frequency | 0.924 | 0.514 | 0.181 | PASS |
| BlockFrequency | 0.719 | 0.595 | 0.911 | PASS |
| CumulativeSums | 0.739 | 0.003 | 0.058 | PASS |
| Runs | 0.224 | 0.437 | 0.911 | PASS |
| LongestRun | 0.455 | 0.955 | 0.008 | PASS |
| Rank | 0.129 | 0.289 | 0.834 | PASS |
| FFT | 0.045 | 0.080 | 0.021 | PASS |
| NonOverlappingTemplate | 0.224 | 0.058 | 0.616 | PASS |
| OverlappingTemplate | 0.867 | 0.455 | 0.129 | PASS |
| Universal | 0.699 | 0.924 | 0.383 | PASS |
| ApproximateEntropy | 0.236 | 0.616 | 0.383 | PASS |
| RandomExcursions | 0.037 | 0.324 | 0.500 | PASS |
| Serial | 0.474 | 0.946 | 0.851 | PASS |
| LinearComplexity | 0.171 | 0.657 | 0.436 | PASS |

To assess the randomness of the encrypted data, we evaluated the proportion of sequences that passed a specific statistical test. Using a significance level of $\alpha = 0.01$ and considering $n = 100$ sequences, we determined the acceptable range of proportions using the confidence interval formula $\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, where $\hat{p} = 1 - \alpha$. If the proportion falls outside of this interval, it suggests evidence of nonrandomness.

For $n = 100$ and $\alpha = 0.01$, the calculated confidence interval is $0.99 \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.99 \pm 0.0298$ (i.e., the proportion should be greater than 0.9602).

Based on Table I, the ciphertext generated by our designed public-key steganography encoder *SE* based on three methods have all successfully passed 15 types of tests in the NIST SP 800-22 suite. This ensures that the stegotext and covertext produced through this ciphertext steganography are indistinguishable, confirming the effectiveness of our approach in maintaining the indistinguishability between stegotext and covertext and the universality of our framework.

### B. Steganalysis Experiments

Although we have proven the security of the public-key steganography framework in Theorem 2, we continue to engage in steganalysis to differentiate between covers (generated by random sampling) and stegotext (generated by steganographic sampling), ensuring the integrity of this research. Steganalysis is a technology used to discern stegotext from covertext, primarily relying on binary classifiers:

$$\mathbf{F}(X) = \begin{cases} 0, & \text{if } \Phi(X) < 0.5 \\ 1, & \text{if } \Phi(X) \geq 0.5, \end{cases} \tag{30}$$

where $\Phi(X) \in [0, 1]$ represents the probability that the input $X$ is covertext ($\mathbf{F} = 0$) or stegotext ($\mathbf{F} = 1$). A false alarm occurs when $X$ is a covertext while $\mathbf{F} = 1$, and a missed detection occurs when $X$ is a stegotext while $\mathbf{F} = 0$. False alarm and missed detection are defined as follows, respectively:

$$P_{\text{FA}} = \Pr\{\mathbf{F}(X) = 1 \mid X \in \mathcal{C}\}, \tag{31}$$

$$P_{\text{MD}} = \Pr\{\mathbf{F}(X) = 0 \mid X \in \mathcal{S}\}. \tag{32}$$

TABLE II

AN EXAMPLE OF COVERTEXT AND STEGOTEXT PAIRS UNDER TRUNCATION PARAMETER $p = 0.95$

| |
|---|
| **Context:**<br>Years later, he would find himself |
| **Covertext:**<br>Years later, he would find himself standing in front of the same old oak tree, but this time it was different. The once-thriving tree had withered and died, its branches bare and lifeless. The man felt a pang of sadness as he gazed upon the remnants of his childhood memories. He remembered the countless hours he spent playing under that very tree, chasing after acorns and watching them fall to the ground. But now, everything seemed so different. The world had moved on without him, leaving behind only shadows of what once was. The man couldn't help but feel lost and alone, like a ghost from a bygone era. As he stood there, mired in thought, a gentle breeze rustled through the leaves, causing them to whisper secrets of the past. The man closed his eyes, letting the memories wash over him like ... |
| **Stegotext:**<br>Years later, he would find himself standing in front of the same building, but it was no longer a place of wonder and curiosity. It had been transformed into a soulless office tower, devoid of any character or charm. He couldn't help but feel a sense of loss as he gazed upon its sterile facade, remembering the countless hours he had spent exploring every corner of that magical place. As he walked away from the building, he realized that his own life had taken on a similar tone. Once full of excitement and possibility, it now felt stifling and predictable. He found himself longing for the freedom and adventure of those early days, when anything seemed possible. But even as he looked back wistfully at the past, he knew that he couldn't turn back the clock. Life had moved forward, and with it, so had he. The world had changed, and he had changed along ... |

TABLE III

STEGANALYSIS RESULTS FOR THREE PUBLIC-KEY STEGANOGRAPHY INSTANCES

| | Steganalyzer | $P_{\mathrm{E}}$ | $P_{\mathrm{FA}}$ | $P_{\mathrm{MD}}$ |
|---|---|---|---|---|
| Instance on P-384 | FCN | 49.68% | 49.70% | 49.66% |
| | R-BiLSTM-C | 50.24% | 50.10% | 50.38% |
| | BiLSTM-Dense | 50.40% | 50.80% | 50.00% |
| Instance on secp256k1 | FCN | 49.88% | 49.75% | 50.01% |
| | R-BiLSTM-C | 50.52% | 50.40% | 50.64% |
| | BiLSTM-Dense | 50.14% | 49.90% | 50.38% |
| Instance on P-256 | FCN | 50.23% | 50.12% | 50.34% |
| | R-BiLSTM-C | 49.45% | 49.60% | 49.30% |
| | BiLSTM-Dense | 49.78% | 49.50% | 50.06% |

Here, $\mathcal{C}$ and $\mathcal{S}$ represent the covertext set and the stegotext set, respectively. Then, the overall performance is determined by the probability of detection error computed from $P_{\mathrm{FA}}$ and $P_{\mathrm{MD}}$ as follows:

$$P_{\mathrm{E}} = \frac{P_{\mathrm{FA}} + P_{\mathrm{MD}}}{2}. \tag{33}$$

We instantiated our public-key steganography framework based on admissible encoding using Discop [23] with Llama-2-7B on P-384, secp256k1, and P-256, respectively. Covertext and stegotext were generated in pairs with identical contexts, and all covertext and stegotext were generated with a truncation parameter of $p = 0.95$. To establish the dataset, we randomly selected various short sentences as contexts and generated 10,000 covers and 10,000 stegotext. The example of covertext and stegotext pairs are shown in Tab. II. We then employed three linguistic steganalyzers, including FCN [40], R-BiLSTM-C [41], and BiLSTM-Dense [42]. The steganalysis experiments were conducted on a dataset of 10,000 samples of stegotext and covertext, divided into training, validation, and test sets in ratios of 3:1:1. The result of steganalysis for our provably secure public-key steganography based on admissible encoding is shown in Tab. III.

The results are presented in Tab. III, which reveals that even under such a large scale, the detection error rates are still close to 50%. The false alarm rate and missed detection rate are also consistently near 50%. These results suggest that it is challenging to distinguish between the distribution of stegotext with secret information and the randomly sampled covertext.

Through the steganalysis of these three instances, we have demonstrated that our framework can be safely deployed on all commonly used curves, greatly increasing the applicability of public-key steganography. Furthermore, due to the regularity property of admissible encoding, in algorithm Alg. 2, the *Point Hiding* step will terminate the loop in $O(1)$ time for any sampled curve point. In other words, our public key steganography can efficiently cover all curve points.

Furthermore, in Appendix A, we list a plethora of elliptic curve well-distributed encoding methods that can be applied within our framework to construct public-key steganography and present the algorithm for hyperelliptic curves with genus larger than 1. Through the tensor exponent function, our scheme can be extended to even work on hyperelliptic curves.

## VI. CONCLUSION

In this paper, we propose a general and complete public-key steganography framework based on admissible encoding which can be employed on all types of curves and utilize all points on the curve. Due to the strict requirements of existing point compression methods on curve parameters, and the inability to establish a surjective mapping from all curve points to a one-dimensional finite field, we consider establishing mappings on two-dimensional or even higher-dimensional finite fields. By utilizing some imperfect but well-distributed encoding techniques through the tensor square (on genus 1 curves) or tensor exponent (on curves with genus larger than 1), we construct the powerful tool called admissible encoding. This encoding forms a surjection onto the set of curve points, and under the premise of constructing suitable sampleable inverse functions, it can decode corresponding curve points from multidimensional finite fields. These properties provide theoretical and algorithmic foundations for achieving provably secure public-key steganography.

The significance of the work described in this paper lies in our achievement of extending almost all provably secure steganography methods to the public-key setting while removing all restrictions on curve parameters and allowing encoding for all points. This provides the most universal interface for deploying related higher-level protocols. Subsequent researchers can utilize our work to consider the implementation and deployment of covert protocols on social networks, such as secret sharing or group key agreement, which is a very interesting topic.

Moreover, we aim to broaden the scope of steganography by incorporating additional communication protocols into covert application scenarios [43]. Looking ahead, we envision the

potential to create a parallel environment within large-scale applications. In this parallel world, we would have communication tools akin to those in the real world, yet remain undetectable to external observers.

**Algorithm 7** Public-key Steganography Encoder (SE) Extended to Hyperelliptic Curves

**INPUT:** $\mathcal{E},\ m,\ (p,\ E(\mathbb{F}_p),\ g,\ q,\ PK),\ (f,\ \mathcal{I})$
**OUTPUT:** $s_1, s_2, \cdots, s_*$

## *Key Deriving*
Pick $a \in [0,\ n-1]$ at random;
$P = a \cdot g$
$K = H(a \cdot PK)$
## *Point Hiding*
**while** True **do**
  Pick $v_1\ \cdots\ v_d \in \mathbb{F}_p$ at random;
  $D \leftarrow \mathcal{I}(P - f(v_1) - f(v_2) - \cdots - f(v_d))$;
  pick $i$ uniformly at random in $\#D$;
  $v_{d+1} \leftarrow$ $i$-th element of $D$;
  **if** $u = \varnothing$ **continue**;
  **else break**;
**end while**
## *Bias Eliminating*
Choose $r$-bit redundancy:
Pick $r_1, r_2, \cdots, r_{d+1} \in \left\{0, \ldots, \left\lfloor \frac{2^{k+r}-v_i}{q} \right\rfloor\right\}$
$\tilde{v}_i = v_i + r_1 p,\ \forall\ i \in \{1 \cdots d+1\}$
## *Final Encoding*
$C_1 = \tilde{v_1} \| \tilde{v_2} \| \cdots \| v_{\tilde{d+1}}$
$C_2 = E_K(K, m)$
$s_1,\ s_2, \cdots,\ s_* = \mathcal{E}(C_1 \| C_2)$

**Algorithm 8** Public-key Steganography Decoder (SD) Extended to Hyperelliptic Curves

**INPUT:** $\mathcal{D},\ s_{1..*},\ (p,\ E(\mathbb{F}_p),\ g,\ n,\ SK),\ f$
**OUTPUT:** $m$

Split $C = \mathcal{D}(s_{1..*})$ into $C_1,\ C_2$;
$(\tilde{v}_1,\ \tilde{v}_2,\ \cdots,\ \tilde{v}_{d+1}) = C_1$
$v_i = \tilde{v}_i\ mod\ p,\ \forall\ i \in \{1, \cdots, d+1\}$
$P = f(v_1) + f(v_2) + \cdots + f(v_{d+1})$
$K = H(SK \cdot P)$
$m = D_K(K, C_2)$

## APPENDIX

### A. Generalization to Hyperelliptic Curves

The previous public-key steganography encoder (SE), as defined in Algorithm 2, and the public-key steganography decoder (SD), as defined in Algorithm 3, are specifically designed for deployment on curves of genus 1. To adapt these algorithms for use on hyperelliptic curves, which have a higher genus, it is necessary to transition from employing the tensor square function to utilizing the tensor exponent function.

Let the curve $E(\mathbb{F}_p)$ of genus $d$ defined on finite field $\mathbb{F}_p$ and $f$ is its computable $B$-well-distributed encoding. Other definitions are similar to those mentioned previously. We list below the provably secure public-key steganography encoder (SE, Alg. 7) and decoder (SD, Alg. 8) algorithms that can be deployed on hyperelliptic curves.

According to Lemma 4, we can prove the admissibility of the tensor exponent function in a manner similar to Lemma 1. With the remaining parts unchanged, our public-key steganography scheme can withstand CHA.

The algorithm is largely similar to the one described above, with the sole distinction being in the Point Hiding part, where we utilize a tensor square construction of admissible encoding with $s = d + 1$. As a result, the first $d$ finite field elements are generated by random sampling, and the last element is obtained through sampling using the sampleable inverse function of $f$. Hence, the length of $C_1$ is a bit sequence of $d + 1$ segments.

We have listed a plethora of elliptic curve well-distributed encoding method in Tab. IV that can be applied within our framework to construct public-key steganography. Through the tensor exponent function, our scheme can be extended to even work on hyperelliptic curves.

### B. List of Key Lemmas

*Lemma 3 (Hasse Bound)* [44] For curve $E(\mathbb{F}_p)$ of genus 1 defined over finite field $\mathbb{F}_p$, we have:

$$|\#E(\mathbb{F}_p) - p - 1| \le 2\sqrt{p} \tag{34}$$

*Lemma 4 ([30], Corollary 4):* If $f : \mathbb{F}_p \to E(\mathbb{F}_p)$ is a $B$-well-distributed encoding into a curve $E(\mathbb{F}_p)$, then the statistical distance between the distribution defined by a tensor exponent function $f^{\otimes s}$ on $E(\mathbb{F}_p)$ and the uniform distribution is bounded as:

$$\sum_{D \in E(\mathbb{F}_p)} \left| \frac{N_s(D)}{p^s} - \frac{1}{\#E(\mathbb{F}_p)} \right| \le \frac{B^s}{p^{s/2}} \sqrt{\#E(\mathbb{F}_p)}, \tag{35}$$

*Lemma 5 ([30], Theorem 8):* Let $f$ be Icart's encoding function (23). For any nontrivial character $\chi$ of $E(\mathbb{F}_p)$, the character sum $S_f(\chi)$ given by (6) satisfies:

$$\left|S_f(\chi)\right| \le 12\sqrt{p} + 3. \tag{36}$$

In other words, Icart's encoding is a $(12 + 3p^{-\frac{1}{2}})$-well-distributed encoding.

*Lemma 6: [[31], SectionV]* Let $f$ be SW encoding function (26). For any nontrivial character $\chi$ of $E(\mathbb{F}_p)$, the character sum $S_f(\chi)$ given by (6) satisfies:

$$\left|S_f(\chi)\right| \le 62\sqrt{p} + O(1). \tag{37}$$

In other words, Icart's encoding is a $(62 + O\left(p^{-\frac{1}{2}}\right))$-well-distributed encoding.

*Lemma 7 ([30], Theorem 15):* Let $f$ be the SWU encoding function (28). For any nontrivial character $\chi$ of $E(\mathbb{F}_p)$, the character sum $S_f(\chi)$ given by (6) satisfies:

$$\left|S_f(\chi)\right| \le 52\sqrt{p} + 151. \tag{38}$$

In other words, SWU encoding is a $\left(52 + 151p^{-\frac{1}{2}}\right)$-well-distributed encoding.

TABLE IV
KNOWN DETERMINISTIC WELL-DISTRIBUTED ENCODINGS TO COMMONLY USED ELLIPTIC CURVES AND HYPERELLIPTIC CURVES

| char. | curve equation | genus | encoding | conditions on $p$ |
|---|---|---|---|---|
| $\neq 2,3$ | $y^2 = x^3 + ax + b$ | 1 | Skalba [45] | – |
| $\neq 2,3$ | $y^2 = x^3 + ax + b$ | 1 | SWU [32] | $p \equiv 3 \mod 4$ |
| $\neq 2,3$ | $y^2 = x^{2g+1} + ax + b$ | $g$ | Ulas [37] | – |
| $\neq 2,3$ | $y^2 = x^{2g+1} + a_1 x^{2g-1} + \cdots + a_g x$ | $g$ | FT [46] | $p \equiv 3 \mod 4$ |
| any | any | 1 | SW [36] | – |
| 3 | $y^2 = x^3 + ax^2 + b$ | 1 | Brier *et al.* [32] | – |
| $\neq 2,3$ | $y^2 = x^3 + ax + b$ | 1 | Icart [29] | $p = 2 \mod 3$ |
| $\neq 2,3$ | $x^3 + y^3 + 1 = 3dxy$ | 1 | F [47] & KLR [48] | $p = 2 \mod 3$ |
| $\neq 2,3$ | $x^3 + (y + c)(3x + 2a + 2b/y) = 0$ | 2 | KLR [48] | $p \equiv 2 \mod 3$ |
| $\neq 2,3$ | $y^2 = x^{2d} + x^d + a$ | $d - 1$ | KLR [48] | $(d, p - 1) = 1$ |
| $\neq 2,3$ | $y^2 = p^{(d)}_{a,b}(x)$ | $\frac{d-1}{2}$ | KLR [48] | $p \equiv 2 \mod 3, (d, p - 1) = 1$ |
| 2 | $y^2 + y = p^{(d)}_{a,b}(x)$ | $\frac{d-1}{2}$ | KLR [48] | $p = 2 \mod 3$ |
| 2 | $y^2 + xy = x^3 + ax^2 + b$ | 1 | Icart [29] | $p = 2 \mod 3$ |

## REFERENCES


[1] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 474–481, May 1998.

[2] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1075–1083, Aug. 1999.

[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Morgan Kaufmann, 2007.

[4] G. J. Simmons, "The Prisoners' problem and the subliminal channel," in *Advances in Cryptology*. Cham, Switzerland: Springer, Jan. 1984, pp. 51–67.

[5] Z.-L. Yang, X.-Q. Guo, Z.-M. Chen, Y.-F. Huang, and Y.-J. Zhang, "RNN-Stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 1280–1295, 2018.

[6] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[7] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[8] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 781–783, 2006.

[9] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 243–255, Feb. 2015.

[10] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*. Cham, Switzerland: Springer, 2010, pp. 161–177.

[11] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017.

[12] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, Sep. 2018.

[13] C. Cachin, "An information-theoretic model for steganography," in *Proc. Int. Workshop Inf. Hiding*. Cham, Switzerland: Springer, Apr. 1998, pp. 306–318.

[14] N. J. Hopper, J. Langford, and L. Von Ahn, "Provably secure steganography," in *Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2002, pp. 77–92.

[15] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[16] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," 2016, *arXiv:1602.02410*.

[17] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3617–3621.

[18] Z. Ziegler, Y. Deng, and A. Rush, "Neural linguistic steganography," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1210–1215.

[19] K. Chen, H. Zhou, H. Zhao, D. Chen, W. Zhang, and N. Yu, "Distribution-preserving steganography based on text-to-speech generative models," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 5, pp. 3343–3356, Sep. 2022.

[20] S. Zhang, Z. Yang, J. Yang, and Y. Huang, "Provably secure generative linguistic steganography," 2021, *arXiv:2106.02011*.

[21] G. Kaptchuk, T. M. Jois, M. Green, and A. D. Rubin, "Meteor: Cryptographically secure steganography for realistic distributions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 1529–1548.

[22] C. S. d. Witt, S. Sokota, J. Z. Kolter, J. Foerster, and M. Strohmeier, "Perfectly secure steganography using minimum entropy coupling," in *Proc. 11th Int. Conf. Learn. Represent.*, Jan. 2022.

[23] J. Ding, K. Chen, Y. Wang, N. Zhao, W. Zhang, and N. Yu, "Discop: Provably secure steganography in practice based on 'distribution copies,"' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 2238–2255.

[24] L. Von Ahn and N. J. Hopper, "Public-key steganography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland. Berlin, Germany: Springer, May 2004, pp. 323–341.

[25] X. Zhang, K. Chen, J. Ding, Y. Yang, W. Zhang, and N. Yu, "Provably secure public-key steganography based on elliptic curve cryptography," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3148–3163, 2024.

[26] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, 2001, pp. 213–229.

[27] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Proc. Int. Conf. theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, Jan. 2002, pp. 466–481.

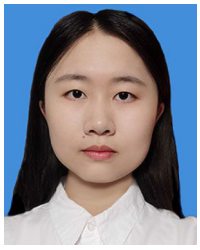
[28] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2003, pp. 416–432.

[29] T. Icart, "How to hash into elliptic curves," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, Jan. 2009, pp. 303–316.

[30] R. R. Farashahi, P.-A. Fouque, I. E. Shparlinski, M. Tibouchi, and J. F. Voloch, "Indifferentiable deterministic hashing to elliptic and hyperelliptic curves," *Math. Comput.*, vol. 82, no. 281, pp. 491–512, Apr. 2012.
[31] P.-A. Fouque and M. Tibouchi, "Indifferentiable hashing to Barreto-Naehrig curves," in *Proc. 2nd Int. Conf. Cryptol. Inf. Secur. Latin Amer.*, Santiago, Chile. Cham, Switzerland: Springer, Jan. 2012, pp. 1–17.
[32] É. Brier, J.-S. Coron, T. Icart, D. Madore, H. Randriam, and M. Tibouchi, "Efficient indifferentiable hashing into ordinary elliptic curves," in *Proc. 30th Annu. Cryptol. Conf.* Cham, Switzerland: Springer, Jan. 2010, pp. 237–254.
[33] S. Katzenbeisser and F. A. Petitcolas, "Defining security in steganographic systems," *Proc. SPIE*, vol. 4675, pp. 50–56, Apr. 2002.
[34] K. Yang, K. Chen, W. Zhang, and N. Yu, "Provably secure generative steganography based on autoregressive model," in *Proc. Int. Workshop Digit. Watermarking*. Cham, Switzerland: Springer, 2018, pp. 55–68.
[35] N. Hopper, "On steganographic chosen covertext security," in *Proc. 32nd Int. Colloq. (ICALP) Automata, Lang. Program.*, Lisbon, Portugal. Berlin, Germany: Springer, Jul. 2005, pp. 311–323.
[36] A. Shallue and C. E. v. d. Woestijne, "Construction of rational points on elliptic curves over finite fields," in *Proc. 7th Int. Algorithmic Number Theory Symp.* Berlin, Germany: Springer, Jan. 2006, pp. 510–524.
[37] M. Ulas, "Rational points on certain hyperelliptic curves over finite fields," 2007, *arXiv:0706.1448*.
[38] V. Shoup, "A new polynomial factorization algorithm and its implementation," *J. Symbolic Comput.*, vol. 20, no. 4, pp. 363–397, Oct. 1995.
[39] P. S. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Proc. Int. Workshop Sel. Areas Cryptogr.* Cham, Switzerland: Springer, 2005, pp. 319–331.
[40] Z. Yang, Y. Huang, and Y.-J. Zhang, "A fast and efficient text steganalysis method," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 627–631, Apr. 2019.
[41] Y. Niu, J. Wen, P. Zhong, and Y. Xue, "A hybrid R-BILSTM-C neural network based text steganalysis," *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1907–1911, Dec. 2019.
[42] H. Yang, Y. Bao, Z. Yang, S. Liu, Y. Huang, and S. Jiao, "Linguistic steganalysis via densely connected LSTM with feature pyramid," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2020, pp. 5–10.
[43] S. Berndt and M. Liśkiewicz, "On the universal steganography of optimal rate," *Inf. Comput.*, vol. 275, Dec. 2020, Art. no. 104632.
[44] H. Niederreiter and C. Xing, *Algebraic Geometry in Coding Theory and Cryptography*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[45] M. Skałba, "Points on elliptic curves over finite fields," *Acta Arithmetica*, vol. 117, no. 3, pp. 293–301, 2005.
[46] P.-A. Fouque and M. Tibouchi, "Deterministic encoding and hashing to odd hyperelliptic curves," in *Proc. Int. Conf. Pairing-Based Cryptography*. Cham, Switzerland: Springer, Jan. 2010, pp. 265–277.
[47] R. R. Farashahi, "Hashing into Hessian curves," in *Proc. Int. Conf. Cryptol. Afr.*, vol. 3, Jan. 2014, p. 139.
[48] J.-G. Kammerer, R. Lercier, and G. Renault, "Encoding points on hyperelliptic curves over finite fields in deterministic polynomial time," in *Proc. Int. Conf. Pairing-Based Cryptography*. Cham, Switzerland: Springer, May 2010, pp. 278–297.

**Xin Zhang** received the B.S. degree from the University of Science and Technology of China (USTC) in 2022, where he is currently pursuing the Ph.D. degree in engineering. His research interests include information hiding, applied cryptography, and deep learning.

**Kejiang Chen** (Member, IEEE) received the B.Eng. degree from Shanghai University (SHU) in 2015 and the Ph.D. degree from the University of Science and Technology of China (USTC) in 2020. He is currently an Associate Professor with USTC. His research interests include information hiding, image processing, and deep learning.



**Na Zhao** received the B.S. degree from Zhengzhou University (ZZU) in 2017. She is currently pursuing the Ph.D. degree with the University of Science and Technology of China. Her research interests include information hiding and deep learning security.

**Weiming Zhang** (Member, IEEE) received the M.S. and Ph.D. degrees from Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.

**Nenghai Yu** (Member, IEEE) received the B.S. degree from Nanjing University of Posts and Telecommunications in 1987, the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China in 2004. He is currently a Professor with the University of Science and Technology of China. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.