

Confidential Error-Correcting Codes

Tianrui Wang¹ Anyu Wang^{2,3,4†} Tianshuo Cong^{5,6†}

Delong Ran¹ Xiaoyun Wang^{2,3,4,7,8}

¹*Institute for Network Sciences and Cyberspace, Tsinghua University*

²*Institute for Advanced Study, Tsinghua University* ³*Zhongguancun Laboratory, Beijing, China*

⁴*State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University*

⁵*School of Cryptologic Science and Engineering, Shandong University*

⁶*Shandong Key Laboratory of Artificial Intelligence Security, Shandong University*

⁷*National Financial Cryptography Research Center, Beijing, China*

⁸*Shandong Institute of Blockchain, Shandong, China*

Abstract

We introduce Confidential Error-correcting Codes (CECs), a new cryptographic primitive that enables reliable decoding from perturbed codewords while preserving message confidentiality. We formalize this primitive through three security properties: confidentiality, robustness, and soundness. To realize this goal, we develop a general framework for constructing CECs based on a Randomness-Security Separation (RSS) structure, which first encodes a nonce to introduce randomness and then masks the encoded payload with a pseudorandom function. Based on our RSS framework, we instantiate CECs, called Conc-CEC, using concatenated Reed-Muller and Reed-Solomon codes. The resulting Conc-CEC admits provable guarantees for all three properties. Notably, compared to PRCs (CRYPTO’24), another recent cryptographic primitive with error-correcting capability, CECs achieve significantly stronger concrete security guarantees under practical parameter settings. As a primary application of CECs, we further present CECMark, a watermarking scheme for AI-generated images. Our experiments on real-world generative image models show that CECMark consistently outperforms state-of-the-art watermarking methods, including Gaussian Shading (CVPR’24), PRCMark (ICLR’25), and T2SMark (NeurIPS’25). In particular, CECMark achieves higher robustness across all 56 evaluated perturbations and ensures a false positive rate below 10^{-5} without introducing side effects. Taken together, our work lays a novel cryptographic foundation for trustworthy watermarking schemes that, for the first time, unify provable security guarantees with strong practical reliability.

1 Introduction

The growth of AI-Generated Content (AIGC) raises misuse concerns [1, 28], making reliable source attribution increasingly important. Against this backdrop, reliable attribution mechanisms have become essential technical building blocks for effective AI governance. For example, the EU AI Act [13]

introduces transparency obligations for AI-generated content, which will become applicable from August 2026 and require certain AI-generated outputs to be marked in a machine-readable and detectable manner. Among the candidate techniques, watermarking [42] is especially appealing because it allows generated content itself to carry provenance-related information while preserving content quality.

However, watermarking schemes face multiple security threats, such as unauthorized extraction of embedded information, malicious perturbation of watermarked content, and attribution errors. Therefore, beyond the common utility requirements (e.g., fidelity), these threats lead to three core security requirements: Confidentiality, Robustness, and Soundness. Despite their importance, existing watermarking schemes still exhibit two fundamental limitations:

- **Neglected Confidentiality.** Robustness and soundness have been widely studied in prior watermarking literature [9], whereas confidentiality has received considerably less attention despite its growing importance in provenance-oriented watermarking systems.
- **Lack of Unified Protection.** Satisfying all three requirements simultaneously is challenging because of the inherent tension between cryptographic randomness and structured redundancy. As a result, SOTA schemes typically optimize only a subset of these properties (see Table 1), and no practical scheme currently achieves all three simultaneously.

Why Confidentiality Matters? The Coalition for Content Provenance and Authenticity (C2PA) [11] emphasizes embedding multi-bit provenance metadata for AIGC authentication and verification. Major AI providers, including OpenAI [32], Claude [10], Azure [30], and Google [17], have adopted mechanisms to embed provenance information into AIGCs. Specifically, the C2PA specification [11] requires provenance information describing who created or modified an asset and how it was changed. Such embedded information may include timestamps, user IDs, digital signatures, or model specifications [19]. While these attributes are essential for enabling

fine-grained provenance and accountability, they may also contain information that should not be accessible to unauthorized parties. For example, user IDs may compromise user privacy and ownership anonymity, while model specifications may enable adversaries to identify and exploit vulnerabilities in specific generation models. This motivates confidentiality as a fundamental requirement for multi-bit watermarking, requiring that an adversary should learn nothing about the embedded payload from the watermarked content.

Why Existing Error-Tolerant Cryptographic Schemes Are Insufficient? A natural question is whether existing cryptographic schemes with error-tolerant capability can already support watermarking. Unfortunately, they typically do not directly fit this setting. Code- and lattice-based public-key schemes, such as BIKE [3], HQC [15], and Kyber [7], introduce noise primarily for security rather than robustness, while conventional combinations of symmetric encryption and error-correcting codes do not jointly provide IND-CPA randomness and perturbation robustness. A detailed analysis is provided in Appendix C.1. A particularly promising cryptographic primitive is Pseudorandom Error-correcting Codes (PRCs) [9], introduced at CRYPTO 2024. Nevertheless, existing PRC constructions [8, 9, 16, 19] remain difficult to instantiate securely under practical parameters. For example, [39] shows that part of the secret key of the LDPC-based PRC construction in [9] can be recovered, thereby compromising the confidentiality of the embedded watermark. We will present a comprehensive analysis of the limitations of all currently known PRC instances in Appendix L.

1.1 Our Contributions

To bridge these gaps, we introduce *Confidential Error-Correcting Codes (CECs)*, a new cryptographic primitive designed to capture message confidentiality, perturbation robustness, and sound decoding within a unified coding framework. Figure 1 summarizes the organization of our paper.

Definition of CECs. CECs are defined to satisfy three properties: (i) *Confidentiality* requires that any probabilistic polynomial-time adversary has only a negligible advantage in the IND-CPA game, ensuring that the codewords reveal no information about the embedded plaintext information. (ii) *Robustness* corresponds to the error-correcting capability, guaranteeing correct decryption even when the ciphertext is subject to bounded perturbation. (iii) *Soundness* requires that a random vector be successfully decoded to a valid message only with negligible probability. The formal definition of CECs will be provided in Section 3.1.

Construction of CECs. To systematically generate the codewords of CECs, we develop a generic construction framework based on a *Randomness-Security Separation (RSS)* structure (see Figure 2). This design creates a clean separation of roles: Confidentiality is achieved solely from the security of Pseudorandom Functions (PRFs), without requiring any

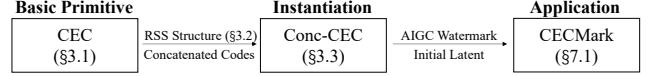


Figure 1: Overview of the paper organization.

cryptographic indistinguishability property from the underlying error-correcting codes. That said, such a separation allows the error-correcting component to focus on Robustness and Soundness, while security is ensured independently.

Instantiation of CECs. In Section 3.3, based on the RSS structure, we present Conc-CEC, which instantiates the error-correcting component using concatenated Reed-Muller and Reed-Solomon (RM and RS) codes. This concatenated design provides strong Robustness, supported by a layered Decoding Failure Rate (DFR) analysis in theory. Meanwhile, we will compare this with duplicated codes in Section 8.1 to highlight the advantage of our design choice. Moreover, the restricted decoding region reduces the probability of false acceptance, thereby contributing to Soundness. Above all, Conc-CEC admits provable security for the following reasons.

- **Provable Confidentiality:** Since the encoded payload is protected only through PRF-based masking, confidentiality directly reduces to the security of the PRF, independent of the error-correcting code. Section 4 will provide a detailed theoretical analysis of the provable confidentiality by using the IND-CPA game (Game 2).
- **Provable Robustness:** Robustness is delegated to the error-correcting component, which ensures reliable recovery under perturbations and noise. In Section 5, we will analyze robustness by deriving the DFR of the concatenated code under a given noise rate. The overall DFR is characterized through a layered analysis of the RM and RS components.
- **Provable Soundness:** Soundness is ensured by the bounded decoding radius of the code, which restricts decoding to a small neighborhood of valid codewords, making it unlikely for arbitrary inputs to be mapped to valid messages. The formal proof is given in Section 6.

CEC-based Watermarking Schemes. In Section 7, we propose CECMark, a novel watermarking scheme for generative image models based on CECs. To the best of our knowledge, CECMark is the first watermarking scheme to unify provable security guarantees with strong practical reliability. CECMark embeds the codewords of Conc-CEC into the signs of latent of generative models restricted to those with large absolute values, enabling both security guarantees and reliable detection. Theoretically, CECMark achieves a 2^{64} -level randomness and a 2^{256} -level confidentiality by selecting a 64-bit nonce and instantiating the underlying PRF with 256-bit security. This provides substantially stronger provable security than GS, PRCMark, and T2SMark, whose confidentiality levels are all

below 2^{28} (Section 7.4). For soundness, CECMark maintains a false positive rate below 10^{-5} without introducing side effects (Section 7.6). We further empirically evaluate CECMark on Stable Diffusion 2.1 [33]. Section 7.3 demonstrates that CECMark preserves high image quality. For robustness, Section 7.5 shows that CECMark consistently achieves higher detection accuracy than prior watermark schemes across all 56 evaluated image perturbations. These results demonstrate the practical applicability of CECs in trustworthy watermarking systems. More generally, CECs provide a principled foundation for developing cryptographically grounded and reliably deployable watermarking solutions.

2 Background

2.1 Basic Cryptographic Definitions

We review the cryptographic definitions used by CECs, beginning with PRFs [6] in Definition 1.

Definition 1 (Pseudorandom Function). *A pseudorandom function (PRF) is a function F defined over a keyspace $\mathcal{K}$, input space $\mathcal{X}$, and output space $\mathcal{Y}$: $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$. We have $\mathcal{K} = \mathbb{F}_2^{K(\lambda)}$, $\mathcal{X} = \mathbb{F}_2^{I(\lambda)}$, and $\mathcal{Y} = \mathbb{F}_2^{O(\lambda)}$, where λ is the security parameter and $K(\cdot), I(\cdot), O(\cdot)$ are polynomial functions of λ . The function F is a PRF if it satisfies the following properties:*

1. **Efficient Computability:** *For any $k \in \mathcal{K}$ and $x \in \mathcal{X}$, the value $F(k, x)$ can be computed in polynomial time.*
2. **Pseudorandomness:** *It is computationally infeasible to distinguish $F(k, \cdot)$ from a truly random function $R : \mathcal{X} \rightarrow \mathcal{Y}$. Formally, for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_{\text{GUESS}})$, where $\mathcal{A}_{\text{GUESS}}^{\mathcal{O}}$ is given oracle access to $\mathcal{O}$ and can make an adaptive polynomial number of queries to $\mathcal{O}$, we require the advantage*

$$\text{Adv}_{\mathcal{A}}^{\text{PRF}}(\lambda) := \left| \Pr[\text{Game}_{\mathcal{A}}^{\text{PRF}}(\lambda) = 1] - \frac{1}{2} \right|,$$

is negligible in λ , where the game is defined in Game 1.

Game 1 $\text{Game}_{\mathcal{A}}^{\text{PRF}}(\lambda)$

1. $b \xleftarrow{\$} \{0, 1\}$
2. $k \xleftarrow{\$} \mathcal{K}, \quad R \xleftarrow{\$} \{\mathcal{X} \rightarrow \mathcal{Y}\}$
3. $\mathcal{O}(\cdot) := \begin{cases} F(k, \cdot) & \text{if } b = 0, \\ R(\cdot) & \text{if } b = 1 \end{cases}$
4. $b' \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda)$
5. return $(b' = b)$

Proving the existence of PRFs remains a fundamental challenge as it implies $P \neq NP$. Nevertheless, PRFs are widely trusted and can be constructed from one-way functions or Pseudorandom Generators (PRGs), such as the NIST-specified DRBG [4] instantiated with AES [12] or

HMAC [26]. Furthermore, stream ciphers such as RC4 [25], ISAAC [35], and ChaCha20 [5] can also be converted into PRFs. We instantiate the PRF with ChaCha20 [5] for its security and broad adoption. We next define IND-CPA security in Definition 2.

Definition 2 (IND-CPA Security). *Let $(\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ be an encryption scheme. We say it is IND-CPA secure if for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_{\text{CHOOSE}}, \mathcal{A}_{\text{GUESS}})$, where $\mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}$ and $\mathcal{A}_{\text{GUESS}}^{\mathcal{O}}$ are given oracle access to $\mathcal{O}$ and can make an adaptive polynomial number of queries to $\mathcal{O}$, the advantage*

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) := \left| \Pr[\text{Game}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) = 1] - \frac{1}{2} \right|$$

is negligible in λ , where the game is defined in Game 2.

Game 2 $\text{Game}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$

1. $\text{sk} \xleftarrow{\$} \mathbb{F}_2^{\lambda_1}$
2. $\mathcal{O}(\cdot) := \text{Encrypt}(1^\lambda, \text{sk}, \cdot)$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$
4. $b \xleftarrow{\$} \{0, 1\}$
5. $\mathbf{c}^* \leftarrow \text{Encrypt}(1^\lambda, \text{sk}, \mathbf{m}_b)$
6. $b' \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, \mathbf{c}^*)$
7. return $(b' = b)$

2.2 Coding Theory

Basic Notations. Let $\mathbb{F}_q$ denote a finite field with q elements. For a vector $\mathbf{x} \in \mathbb{F}_q^n$, let $w_H(\mathbf{x})$ denote its Hamming weight. For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, the Hamming distance between them is $d(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} - \mathbf{y})$. In the following, we introduce some basic concepts related to error-correcting codes.

Definition 3 (Minimum Distance). *The minimum distance of a code $\mathcal{C}$ is $d_{\min}(\mathcal{C}) = \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\}$.*

Definition 4 (Linear Codes). *A q -ary linear code $\mathcal{C}$ of length n is a linear subspace of $\mathbb{F}_q^n$. If $\mathcal{C}$ has dimension k , then $\mathcal{C}$ is called a $[n, k]$ or $[n, k]_q$ code. If the minimum distance of $\mathcal{C}$ is $d_{\min}$, we then describe it as a $[n, k, d_{\min}]$ or $[n, k, d_{\min}]_q$ code.*

Definition 5 (Reed-Muller Codes). *A Reed-Muller code $\text{RM}(r, m)$ with positive integer m and $0 \leq r \leq m$ has parameters $[n, k, d_{\min}] = [2^m, \sum_{i=0}^r \binom{m}{i}, 2^{m-r}]$, and duplicated RM codes are constructed by repeating each codeword multiple times.*

Definition 6 (Reed-Solomon Codes). *Let p be a prime number and q be a power of p . A Reed-Solomon code $\text{RS}[n, k, d_{\min}]$ with symbols in $\mathbb{F}_q$ has block length $n = q - 1$, dimension k , and minimum distance $d_{\min} = n - k + 1$. A shortened Reed-Solomon code is obtained from an $\text{RS}[n, k, d_{\min}]$ code by fixing s message symbols and removing the corresponding s codeword positions. The resulting code has parameters $\text{RS}[n - s, k - s, d_{\min}]$, while preserving the minimum distance.*

Table 1: Comparison of SOTA watermarking schemes with respect to the provable security properties. ● indicates that the property is satisfied, ○ denotes that it is not achieved, and ◐ signifies that the scheme is provably secure only in theory, but remains insecure in practical deployments.

Watermarking Schemes	Provable Security Properties		
	Confidentiality	Robustness	Soundness
Gaussian Shading (GS) [41]	○	●	◐
PRCMark [19]	◐	●	●
T2SMark [40]	◐	●	●
CECMark (Ours)	● ^(a)	● ^(b)	● ^(c)

^(a) Confidentiality is theoretically established but remains practically vulnerable in PRCMark and T2SMark (see Sections 4 and 7.4).

^(b) Robustness follows the order CECMark > PRCMark > T2SMark ≈ GS (see Sections 5 and 7.5).

^(c) Soundness is inherently ensured in CECMark, whereas PRCMark and T2SMark incur additional side effects (see Sections 6 and 7.6).

Concatenated codes are a powerful class of error-correcting codes that leverage a multi-stage structure to achieve better decoding performance. We present the definition of concatenated codes in Definition 7 and describe their encoding procedure in Appendix D.

Definition 7 (Concatenated Codes). *A concatenated code over $\mathbb{F}_2$ consists of an external code C_e with parameters $[n_e, k_e, d_e]$ over $\mathbb{F}_q$ and an internal code C_i with parameters $[n_i, k_i, d_i]$ over $\mathbb{F}_2$, where $q = 2^{k_i}$. The resulting concatenated code is a binary code with parameters $[N = n_e n_i, K = k_e k_i, D \geq d_e d_i]$.*

2.3 Generative Image Models (GIMs)

A Generative Image Model (GIM) is a probabilistic algorithm that takes a textual prompt as input and outputs an image. We focus on the most widely used latent diffusion-based GIMs, which consist of a de-noising model U-net and an autoencoder pair (E-net, D-net). The generation and inversion processes are illustrated in Figure 10. In the generation process, the model samples an initial latent $\mathbf{z}_T$ from a Gaussian distribution. The U-net then progressively de-noises this latent into $\mathbf{z}_0$ under the guidance of the textual prompt over T steps. The decoder D-net finally maps $\mathbf{z}_0$ to the image space and produces the generated image. In the inversion process, an input image is first encoded by E-net into a proxy latent $\hat{\mathbf{z}}_0$. The U-net is then used to approximately invert the de-noising trajectory and obtain a proxy initial latent $\hat{\mathbf{z}}_T$. It allows information embedded in the initial latent $\mathbf{z}_T$ to be approximately recovered from the generated image through generating $\hat{\mathbf{z}}_T$.

2.4 In-processing Watermarking Schemes

In-processing watermarking schemes typically leverage the initial latent $\mathbf{z}_T$ to embed watermarks. We review the embedding and extraction methods of SOTA schemes, specifically Gaussian Shading [41], PRCMark [19], and T2SMark [40].

Table 1 summarizes their provable security properties and compares them with CECMark.

Gaussian Shading (GS) [41]. In the GS scheme, the message is first encoded via duplicated codes and subsequently masked by a vector generated by the secret stream key. The resulting vector is then embedded into the signs of the latent. The extraction process operates inversely: the received vector is first de-masked, and then decoded by the duplicated codes. However, due to the lack of randomness, identical messages always produce identical outputs, which fails to provide confidentiality. Furthermore, the reliance on simple duplicated codes results in limited robustness and soundness.

PRCMark [19]. PRCMark is a practical instantiation based on the PRC primitive [9]. For watermark embedding, it directly adopts the encode algorithm from the LDPC-PRC in [9] to generate binary codewords, which are then mapped to the signs of the initial latent. For watermark extraction, it offers a decode function that employs a Belief Propagation (BP) algorithm to recover the message. While the scheme requires low LDPC sparsity to maintain robustness, this requirement introduces a vulnerability to polynomial-time key recovery attacks that ultimately compromise confidentiality [39].

T2SMark [40]. T2SMark adopts a two-stage design for embedding. The first stage of T2SMark involves encrypting an encoded session key with a master key, while the second stage encrypts the encoded message using that session key. To ensure robustness, T2SMark utilizes duplicated codes in its encoding process, following the same approach as GS. In terms of embedding strategy, while GS and PRCMark directly assign the vector bits to the signs of the latent space, T2SMark introduces an optimization that restricts watermark embedding to positions with high magnitude. However, the restricted 16-bit length of both the session and master keys renders the scheme susceptible to straightforward key recovery. Additionally, the use of duplicated codes leads to insufficient robustness and soundness.

3 Confidential Error-correcting Codes

In this section, we propose a new primitive called *Confidential Error-Correcting Codes (CECs)*. We first introduce the definition of CECs. Next, we propose a construction framework with a randomness-security separation structure. Finally, we instantiate CECs with concatenated codes.

3.1 Definition of CECs

We formalize CECs through three properties: *confidentiality*, *robustness*, and *soundness*.

Definition 8 (CECs). Let Σ be a fixed alphabet, $\mathcal{E} : \Sigma^* \rightarrow \Sigma^*$ be a noisy channel modeled by adding a noise vector with a Bernoulli distribution, and $k(\lambda) \in \mathbb{N}$ be the message length under any fixed λ . A CEC is a triple of polynomial-time randomized algorithms (KeyGen, Encrypt, Decrypt) that satisfies

- **(Confidentiality)** For any polynomial-time adversary $\mathcal{A}$, let $\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ denote the advantage of the IND-CPA game. Then

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) \leq \text{negl}(\lambda),$$

where the definition of the IND-CPA game is in Game 2.

- **(Robustness)** For any message $\mathbf{m} \in \Sigma^{k(\lambda)}$, any $\lambda \in \mathbb{N}$, and $\mathbf{x} \leftarrow \text{Encrypt}(1^\lambda, \text{sk}, \mathbf{m})$,

$$\Pr_{\text{sk} \leftarrow \text{KeyGen}(1^\lambda)} [\text{Decrypt}(1^\lambda, \text{sk}, \mathcal{E}(\mathbf{x})) = \mathbf{m}] \geq 1 - \text{negl}(\lambda).$$

- **(Soundness)** For any fixed $\mathbf{c} \in \Sigma^*$,

$$\Pr_{\text{sk} \leftarrow \text{KeyGen}(1^\lambda)} [\text{Decrypt}(1^\lambda, \text{sk}, \mathbf{c}) = \perp] \geq 1 - \text{negl}(\lambda).$$

Overall, CECs can be viewed as a specialized symmetric-key encryption scheme equipped with error-correcting capability. To ensure robustness, this primitive inherently supports perturbations. Consequently, unlike traditional symmetric-key encryption schemes, CECs operate over a ciphertext space that is larger than their plaintext space.

3.2 Construction of CECs

Overview of the RSS Structure. To construct CECs, we propose a framework featuring a Randomness-Security Separation (RSS) design. A high-level illustration of the RSS framework is shown in Figure 2. This architecture streamlines encryption by decoupling it into two pipelines:

- **Randomness Pipeline:** This pipeline processes a random nonce θ to generate the first ciphertext component $\mathbf{c}_1$ through an “Encode” module, providing the necessary session randomness for the transformation. Note that although a master key is involved here, it only generates a static mask and does not provide security.

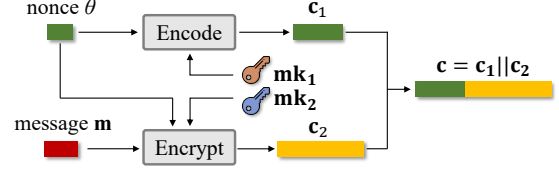


Figure 2: A high-level illustration of the Randomness-Security Separation (RSS) design. The green pipeline provides randomness, while the yellow pipeline ensures security.

- **Security Pipeline:** This branch encrypts the actual message $\mathbf{m}$ into the second component $\mathbf{c}_2$ through an “Encrypt” module. Here, confidentiality is enforced by a mask derived from the master key and the nonce. This pipeline can meet confidentiality requirements because identical plaintexts yield distinct ciphertexts.

Design Benefits of RSS. RSS offers three advantages:

- **Extensibility:** Both the Encode and Encrypt modules in RSS can be replaced with different algorithms. This modularity supports diverse coding schemes, cryptographic primitives, and deployment requirements. The two modules can also be instantiated independently to meet different performance, security, and application needs.
- **Low Overhead:** To minimize overhead, we utilize a short nonce, while security is ensured by a longer master key. This design exploits the gap between adversarial capacity and practical system constraints. Although an adversary may possess substantial computational power (e.g., 2^{64} operations), the number of encryption queries in real-world watermarking schemes is much smaller. By decoupling the randomness space from the security level, we reduce ciphertext overhead without compromising security against realistic adversaries.
- **Functional Separation:** By separating nonce-derived randomness in $\mathbf{c}_1$ from the key-dependent protection in $\mathbf{c}_2$, the RSS design ensures that the overall security is strictly ensured by the master key. This separation allows the choice of codes to focus on error-correcting capability without being constrained by security considerations.

Detailed Construction Workflow. Figure 3 details the workflow of RSS. Given a security parameter λ , our construction is parameterized by $\lambda_1, \lambda_2, k, n_1, n_2, n$, where λ_1 denotes the key length, λ_2 denotes the nonce length, k is the message length, and the output length is $n = n_1 + n_2$. We utilize two error-correcting codes $\mathcal{C}_1 = (\text{Encode}_1, \text{Decode}_1)$ and $\mathcal{C}_2 = (\text{Encode}_2, \text{Decode}_2)$. We also use two functions: $f_1 : \mathbb{F}_2^{\lambda_1} \rightarrow \mathbb{F}_2^{n_1}$ and $f_2 : \mathbb{F}_2^{\lambda_2} \times \mathbb{F}_2^{\lambda_1} \rightarrow \mathbb{F}_2^{n_2}$. Here, f_1 takes the first part of the master key $\mathbf{mk}_1$ as input and expands it into a mask, while f_2 derives a mask from the second part $\mathbf{mk}_2$ together with a nonce. The formal construction pipeline is described as follows.

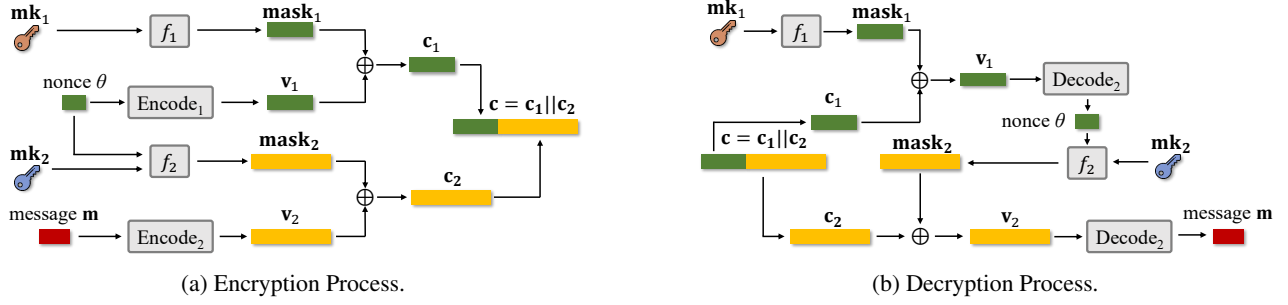


Figure 3: Encryption and decryption procedures of our RSS-based CEC construction framework.

- **KeyGen**(1^λ): Sample $\mathbf{mk}_1, \mathbf{mk}_2 \xleftarrow{\$} \mathbb{F}_2^{\lambda_1} \times \mathbb{F}_2^{\lambda_1}$ and keep the master key $\mathbf{mk} \leftarrow (\mathbf{mk}_1, \mathbf{mk}_2)$ private.
- **Encrypt**($1^\lambda, \mathbf{mk}, \mathbf{m}$): To encrypt a message $\mathbf{m} \in \mathbb{F}_2^k$, first parse $(\mathbf{mk}_1, \mathbf{mk}_2) \leftarrow \mathbf{mk}$. Sample a nonce $\theta \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$, calculate $\mathbf{mask}_1 \leftarrow f_1(\mathbf{mk}_1)$ and $\mathbf{v}_1 \leftarrow \text{Encode}_1(\theta)$. Let $\mathbf{c}_1 \leftarrow \mathbf{v}_1 \oplus \mathbf{mask}_1$. Calculate $\mathbf{mask}_2 \leftarrow f_2(\mathbf{mk}_2, \theta)$. Encode the message $\mathbf{v}_2 \leftarrow \text{Encode}_2(\mathbf{m})$ and calculate $\mathbf{c}_2 \leftarrow \mathbf{v}_2 \oplus \mathbf{mask}_2$. Output $(\mathbf{c}_1, \mathbf{c}_2)$.
- **Decrypt**($1^\lambda, \mathbf{mk}, \mathbf{c}$): Let $(\mathbf{mk}_1, \mathbf{mk}_2) \leftarrow \mathbf{mk}$ and $(\mathbf{c}_1, \mathbf{c}_2) \leftarrow \mathbf{c}$. Calculate $\mathbf{mask}_1 \leftarrow f_1(\mathbf{mk}_1)$. Decode the nonce $\theta' \leftarrow \text{Decode}_1(\mathbf{c}_1 \oplus \mathbf{mask}_1)$. Calculate $\mathbf{mask}_2' \leftarrow f_2(\mathbf{mk}_2, \theta')$. Decode the message $\mathbf{m}' \leftarrow \text{Decode}_2(\mathbf{c}_2 \oplus \mathbf{mask}_2')$ and output $\mathbf{m}'$.

In the randomness pipeline of **Encrypt**($\cdot$), a uniform random nonce θ is first encoded into $\mathbf{v}_1$ and then masked into $\mathbf{c}_1$. Although $\mathbf{c}_1$ includes a mask vector $\mathbf{mask}_1$, it provides static padding but does not ensure security. To save storage space, the scheme stores only the seed $\mathbf{mk}_1$ and derives $\mathbf{mask}_1$ deterministically from it, which is equivalent to storing $\mathbf{mask}_1$ directly. Due to the linearity of the error-correcting code $\mathcal{C}_1$, the codeword $\mathbf{c}_1$ may leak information about the encoded nonce θ . For example, let $\mathbf{c}_1$ and $\mathbf{c}_1'$ be two codewords generated with the same master key. An adversary can compute

$$\begin{aligned} \text{Decode}_1(\mathbf{c}_1 + \mathbf{c}_1') &= \text{Decode}_1(\mathbf{v}_1 + \mathbf{v}_1') \\ &= \text{Decode}_1(\text{Encode}_1(\theta) + \text{Encode}_1(\theta')) \\ &= \text{Decode}_1(\text{Encode}_1(\theta + \theta')) \\ &= \theta + \theta', \end{aligned}$$

which reveals the relation between these two encoded nonces. This demonstrates that directly encoding sensitive information in this pipeline would violate confidentiality. To avoid such leakage, we only encode a uniform random nonce θ in the randomness pipeline. In the security pipeline of **Encrypt**($\cdot$), the message $\mathbf{m}$ is first encoded as $\text{Encode}_2(\mathbf{m})$ and then masked by a vector $\mathbf{mask}_2 \leftarrow f_2(\mathbf{mk}_2, \theta)$. The confidentiality relies on the hidden master key $\mathbf{mk}_2$, while the nonce θ only provides randomness. As a result, the encoded message $\mathbf{m}$ is protected under the mask.

The workflow of **Decrypt**($\cdot$) follows the reverse procedure of **Encrypt**($\cdot$): the nonce θ' is first decoded, which is then used to decrypt the message $\mathbf{m}'$. If the perturbations are within the error-correcting capability of $\mathcal{C}_1$ and $\mathcal{C}_2$, the decryption will be successful. Notably, in addition to the decoding function, our framework inherently provides a detection function: if either Decode_1 or Decode_2 fails, the detection result is false.

3.3 Conc-CEC: An Instantiation of CECs

Based on the proposed RSS framework, we instantiate its components with specific cryptographic and coding algorithms (e.g., concatenated codes). We call the resulting Concatenated code-based CEC instantiation **Conc-CEC**.

Specifically, **Conc-CEC** introduces concatenated duplicated Reed-Muller and shortened Reed-Solomon (RM-RS) codes for $\mathcal{C}_1, \mathcal{C}_2$ (i.e., to instantiate Encode_1 , Decode_1 , Encode_2 , Decode_2 in Figure 3). The internal code is obtained by duplicating the RM(1,7) code with parameters $[n_i, k_i, d_i] = [128, 8, 64]$. The external code is constructed by shortening an RS code over $\mathbb{F}_{2^8}$. With RM-RS codes, we adopt a natural failure-handling strategy, where the decryption algorithm outputs a failure symbol whenever either Decode_1 or Decode_2 fails. This approach neither affects robustness nor incurs additional overhead.

The motivation for choosing concatenated codes follows the same design principle as HQC [15], a NIST standard for key encapsulation mechanisms, where the security of the construction is independent of the specific choice of error-correcting codes. Consequently, the coding component is solely responsible for optimizing decoding performance. In particular, the target application scenarios of CECs lie in the regime with block length $n \approx 10^4$, consistent with HQC, where RM-RS has been shown to perform well. Therefore, under this principle, RM-RS concatenation is adopted in **Conc-CEC** due to its strong empirical performance.

Meanwhile, we instantiate the PRF (f_1 and f_2 in Figure 3) using ChaCha20 [5], a widely deployed and secure stream cipher. This selection is guided by both security and efficiency considerations.

In the following three sections, we will conduct a theoretical analysis of the security properties of Conc-CEC. It should be emphasized that although the choice of each component is not unique, all subsequent theoretical analyses are conducted with respect to Conc-CEC.

4 Confidentiality Analysis of Conc-CEC

The confidentiality of Conc-CEC is defined by requiring that any adversary achieves only a negligible advantage in the IND-CPA game. In this section, we prove that the confidentiality of Conc-CEC holds when secure pseudorandom functions are employed, regardless of the specific error-correcting codes used.

Theorem 1 (Confidentiality of Conc-CEC). *Let f_1 and f_2 be independent secure pseudorandom functions. Then Conc-CEC satisfies the confidentiality notion in Definition 8. In particular, for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_{\text{CHOOSE}}, \mathcal{A}_{\text{GUESS}})$, where $\mathcal{A}$ makes at most $q = \text{poly}(\lambda)$ queries to the oracle $\mathcal{O}$, $\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ is negligible in λ .*

Proof. The proof of Theorem 1 proceeds via a sequence of hybrid games. We gradually replace the PRFs used in the encryption algorithm with truly random functions. These replacements have a negligible effect on the advantage because of the security of PRFs.

Hybrid Games: The proof includes the following games:

- Game 3 is the IND-CPA game for original Conc-CEC.
- Game 4 is the PRF game for f_1 .
- Game 5 is the PRF game for f_2 .
- Game 6 is the IND-CPA game for modified Conc-CEC, where f_1 is replaced by a random function.
- Game 7 is the IND-CPA game for modified Conc-CEC, where f_1, f_2 are replaced by random functions.

The detailed descriptions of the modified games (Game 6 and Game 7) are deferred to Appendix E.

Real Game: Game 3 corresponds to the real IND-CPA experiment. For simplicity, we adopt the convention of representing logical outcomes as binary values: any successful decision or match, i.e., $b = b'$ is denoted by 1, while a mismatch is denoted by 0.

Game 3 $\text{Game}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ (Original IND-CPA Game)
1. $(\mathbf{mk}_1, \mathbf{mk}_2) \xleftarrow{\\$} \mathbb{F}_2^{\lambda_1} \times \mathbb{F}_2^{\lambda_2}$ 2. $\mathcal{O}(\mathbf{m})$: (a) $\theta \xleftarrow{\\$} \mathbb{F}_2^{\lambda_2}$ (b) $\mathbf{c}_1 \leftarrow \text{Encode}_1(\theta) + f_1(\mathbf{mk}_1)$ (c) $\mathbf{c}_2 \leftarrow \text{Encode}_2(\mathbf{m}) + f_2(\mathbf{mk}_2, \theta)$

(d) return $(\mathbf{c}_1, \mathbf{c}_2)$ 3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$ 4. $b \xleftarrow{\\$} \{0, 1\}$ 5. $\theta^* \xleftarrow{\\$} \mathbb{F}_2^{\lambda_2}$ 6. $\mathbf{c}_1^* \leftarrow \text{Encode}_1(\theta^*) + f_1(\mathbf{mk}_1)$ 7. $\mathbf{c}_2^* \leftarrow \text{Encode}_2(\mathbf{m}_b) + f_2(\mathbf{mk}_2, \theta^*)$ 8. $b' \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, (\mathbf{c}_1^*, \mathbf{c}_2^*))$ 9. return $(b' = b)$

Replace $f_1(\mathbf{mk}_1)$: We next modify Game 3 by replacing the value $f_1(\mathbf{mk}_1)$ with a uniformly random vector. Specifically, in Game 6, the value $f_1(\mathbf{mk}_1)$ is replaced by $\mathbf{r}_1 \xleftarrow{\$} \mathbb{F}_2^{n_1}$. Equivalently, we replace $f_1(\mathbf{mk}_1)$ with the output of a truly random function $R_1 : \mathbb{F}_2^0 \rightarrow \mathbb{F}_2^{n_1}$.

PRF Reduction for f_1 : To justify this replacement between Game 3 and Game 6, we construct a distinguisher $\mathcal{B}_1$ against the pseudorandomness of f_1 . The distinguisher $\mathcal{B}_1$ is given oracle access to $\mathcal{O}_1(\cdot)$, which returns either $f_1(\mathbf{mk}_1)$ or the output of a truly random function R_1 . Instead of directly analyzing the oracle $\mathcal{O}_1$ itself, $\mathcal{B}_1$ simulates the IND-CPA challenger for $\mathcal{A}$ as in Game 4. When $\mathcal{A}$ wins its game, i.e., $\hat{\beta} = \beta$, $\mathcal{B}_1$ guesses $b' = 0$, i.e., $\mathcal{O}_1 = f_1(\mathbf{mk}_1)$.

If $\mathcal{O}_1 = f_1(\mathbf{mk}_1)$, the simulation is same as Game 3. Let

$$p_0 := \Pr[\text{Game 3} = 1] = \Pr[\mathcal{A} \text{ wins Game 3}].$$

If $\mathcal{O}_1 = R_1$, the simulation is same as Game 6. Let

$$p_1 := \Pr[\text{Game 6} = 1] = \Pr[\mathcal{A} \text{ wins Game 6}].$$

By the definition of PRF advantage, $\text{Adv}_{\mathcal{B}_1}^{\text{PRF}}$ can be written as

$$\begin{aligned} & \left| \frac{1}{2} (\Pr[\mathcal{B}_1 \text{ wins} \mid \mathcal{O}_1 = f_1] + \Pr[\mathcal{B}_1 \text{ wins} \mid \mathcal{O}_1 = R_1]) - \frac{1}{2} \right| \\ &= \left| \frac{1}{2} (\Pr[\mathcal{A} \text{ wins Game 3}] + \Pr[\mathcal{A} \text{ loses Game 6}]) - \frac{1}{2} \right| \\ &= \left| \frac{p_0 + 1 - p_1}{2} - \frac{1}{2} \right| = \left| \frac{p_0 - p_1}{2} \right|. \end{aligned}$$

Hence, the difference between Game 3 and Game 6 is bounded by

$$|\Pr[\text{Game 3} = 1] - \Pr[\text{Game 6} = 1]| \leq 2 \text{Adv}_{\mathcal{B}_1}^{\text{PRF}}(\lambda). \quad (1)$$

Replace $f_2(\mathbf{mk}_2, \theta)$: We next replace $f_2(\mathbf{mk}_2, \cdot)$ with a truly random function in Game 7. That is, $f_2(\mathbf{mk}_2, \cdot)$ is replaced by $R_2 : \mathbb{F}_2^{\lambda_2} \rightarrow \mathbb{F}_2^{n_2}$. To ensure consistency, we maintain a table $\mathcal{D}$. For each query on input θ , if $\theta \notin \mathcal{D}$, we sample $\mathcal{D}[\theta] \xleftarrow{\$} \mathbb{F}_2^{n_2}$. The output corresponding to θ is then defined as $\mathcal{D}[\theta]$.

PRF Reduction for f_2 : To justify this replacement between Game 6 and Game 7, we construct a distinguisher $\mathcal{B}_2$ against the pseudorandomness of f_2 . The distinguisher $\mathcal{B}_2$ is given oracle access to $\mathcal{O}_2(\cdot)$, which is either the pseudorandom function $f_2(\mathbf{mk}_2, \cdot)$ with a secret $\mathbf{mk}_2$ or a truly random function $R_2(\cdot)$. $\mathcal{B}_2$ simulates the IND-CPA challenger for $\mathcal{A}$ as in

Game 5. If $\mathcal{O}_2 = f_2(\mathbf{mk}_2, \cdot)$, then the simulation is identical to Game 6. If $\mathcal{O}_2 = R_2(\cdot)$, then the simulation is identical to Game 7.

Let $p_1 := \Pr[\text{Game 6} = 1]$ and $p_2 := \Pr[\text{Game 7} = 1]$. Using an analysis similar to that in the PRF reduction for f_1 , we have

$$\text{Adv}_{\mathcal{B}_2}^{\text{PRF}}(\lambda) = \frac{|p_1 - p_2|}{2}.$$

Hence,

$$|\Pr[\text{Game 6} = 1] - \Pr[\text{Game 7} = 1]| \leq 2\text{Adv}_{\mathcal{B}_2}^{\text{PRF}}(\lambda). \quad (2)$$

Analysis of Game 7: In Game 7, $\mathbf{c}_1$ is masked by a uniform random string, and $\mathbf{c}_2$ is masked by the output of a random function R_2 .

$$\mathbf{c}_1 = \text{Encode}_1(\theta) + r_1, \quad \mathbf{c}_2 = \text{Encode}_2(\mathbf{m}_b) + R_2(\theta).$$

Let Bad denote the event that the nonce θ^* is used in any of the q queries to the encryption oracle. We have $\Pr[\text{Bad}] \leq \frac{q}{2^{\lambda_2}}$. Conditioned on $\neg\text{Bad}$, the value $R_2(\theta^*)$ has never been queried before, and hence is a fresh uniform random string independent of the adversary's view. Moreover, r_1 is also a uniform random string independent of everything else. Therefore, the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ is distributed independently of the challenge bit b . It follows that $\Pr[b' = b \mid \neg\text{Bad}] = \frac{1}{2}$. Consequently, $\Pr[b' = b]$ can be bounded as

$$\begin{aligned} & \Pr[b' = b \mid \neg\text{Bad}] \Pr[\neg\text{Bad}] + \Pr[b' = b \mid \text{Bad}] \Pr[\text{Bad}] \\ & \leq \frac{1}{2} (1 - \Pr[\text{Bad}]) + \Pr[\text{Bad}] \leq \frac{1}{2} + \frac{q}{2^{\lambda_2+1}}. \end{aligned} \quad (3)$$

Combining the results in Eq. (1), Eq. (2), and Eq. (3), we have

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda) \leq 2\text{Adv}_{\mathcal{B}_1}^{\text{PRF}}(\lambda) + 2\text{Adv}_{\mathcal{B}_2}^{\text{PRF}}(\lambda) + \frac{q}{2^{\lambda_2+1}}. \quad (4)$$

Since $q = \text{poly}(\lambda)$, $\lambda_2 = \text{poly}(\lambda)$, and both f_1, f_2 are secure PRFs, the advantage $\text{Adv}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ is negligible in λ , which completes the proof. $\square$

5 Robustness Analysis of Conc-CEC

The robustness of Conc-CEC can be characterized through the Decoding Failure Rate (DFR) of the underlying concatenated codes $\mathcal{C}_1$ and $\mathcal{C}_2$. The DFR of RM-RS codes is given in Theorem 2, based on the analysis of [15].

Theorem 2 (DFR of the concatenated code [15]). *The DFR of the concatenated code using an RS code $[n_e, k_e, d_e]_{\mathbb{F}_{256}}$ as the external code and a duplicated RM code as the internal code can be upper bounded as follows:*

$$\text{DFR} = \sum_{l=\delta_e+1}^{n_e} \binom{n_e}{l} p_l^l (1-p_l)^{n_e-l}, \quad (5)$$

where $d_e = 2\delta_e + 1$ and p_i is the upper bound of the DFR of RM codes, as defined in Theorem 3.

Theorem 3 (DFR of the internal code [15]). *Let p denote the noise rate of the channel. The DFR of a duplicated RM code with dimension $k = 8$ and minimum distance d_i is upper-bounded by p_i , which is computed as*

$$\min \left\{ \sum_{w=d_i/2}^n \mathcal{A}_w p^w (1-p)^{n-w}, 255 \sum_{j=d_i/2}^{d_i} \binom{d_i}{j} p^j (1-p)^{d_i-j} \right\},$$

where

$$\begin{aligned} \mathcal{A}_w = \min \left\{ \binom{n}{w}, \frac{1}{2} 255 \binom{d_i}{d_i/2} \binom{d_i}{w-d_i/2} \right. \\ \left. + 255 \sum_{j=d_i/2+1}^{d_i} \binom{d_i}{j} \binom{d_i}{w-j} \right. \\ \left. + \frac{1}{2} \binom{255}{2} \sum_{j=0}^{d_i/2} \binom{d_i/2}{j}^3 \binom{d_i/2}{w-d_i+j} \right\}. \end{aligned}$$

In Section 8.1, we will provide a concrete DFR analysis of RM-RS codes under practical parameter settings. We will further compare RM-RS codes with alternative constructions, including duplicated codes used in GS and T2SMark. This analysis justifies our choice of RM-RS codes by showing, both theoretically and empirically, that they provide superior robustness through a lower DFR.

6 Soundness Analysis of Conc-CEC

We define the False Positive Rate (FPR) as the probability that a random vector is successfully decrypted. Recall that in Section 3.3, Conc-CEC outputs a failure symbol if either Decode_1 or Decode_2 fails. Therefore, a false positive occurs only when both decoding procedures succeed on a random input. For the underlying RM-RS code, we define the FPR analogously as the probability that a random vector is successfully decoded. We begin by analyzing the FPR of a single RM-RS code.

Theorem 4 (FPR of the RM-RS Code). *Let $\mathcal{C} \subseteq \mathbb{F}_q^{n_e}$ be an RS code $RS[n_e, k_e]$ over $\mathbb{F}_q$, with minimum distance $d_e = n_e - k_e + 1$ and error-correcting radius $t_e = \lfloor \frac{d_e-1}{2} \rfloor$. Let $\mathbf{x} \in \mathbb{F}_2^{n_e}$ be uniformly random. Then the probability that $\mathbf{x}$ is decoded to some codeword in $\mathcal{C}$ under bounded-distance decoding equals*

$$\text{FPR} = \sum_{i=0}^{t_e} \binom{n_e}{i} (q-1)^i q^{k_e-n_e}. \quad (6)$$

Proof. Consider the first-stage decoding of RM. The input $\mathbf{x}$ is divided into n_e equal blocks $\mathbf{x} = (\mathbf{x}^{(1)} \parallel \dots \parallel \mathbf{x}^{(n_e)})$. Let

$$\mathbf{y} = (\mathbf{y}^{(1)} \parallel \dots \parallel \mathbf{y}^{(n_e)}), \quad \mathbf{y}^{(i)} = \text{RM.Decode}(\mathbf{x}^{(i)}),$$

be the output of the RM decoder.

Since $\mathbf{x}$ is uniformly random over $\mathbb{F}_2^{n_e}$ and RM.Decode outputs the closest codeword, each $\mathbf{y}^{(i)}$ is uniformly distributed

over its corresponding symbol set. Interpreting $\mathbf{y}^{(i)}$ as an element of $\mathbb{F}_q$, the resulting vector $\mathbf{y} \in \mathbb{F}_q^{n_e}$ is uniformly random.

Let $\mathcal{C}$ denote the $\text{RS}[n_e, k_e]$ code. The code contains exactly $|\mathcal{C}| = q^{k_e}$ codewords. For any codeword $\mathbf{c} \in \mathcal{C}$, define the Hamming ball of radius t_e as

$$B(\mathbf{c}, t_e) = \{\mathbf{v} \in \mathbb{F}_q^{n_e} : d_H(\mathbf{v}, \mathbf{c}) \leq t_e\}.$$

The size of each such ball is

$$|B(\mathbf{c}, t_e)| = \sum_{i=0}^{t_e} \binom{n_e}{i} (q-1)^i.$$

Since RS codes achieve a minimum distance d_e , the Hamming balls $\{B(\mathbf{c}, t_e)\}_{\mathbf{c} \in \mathcal{C}}$ are pairwise disjoint. Therefore, the total number of vectors in $\mathbb{F}_q^{n_e}$ that can be successfully decoded equals

$$|\mathcal{C}| \cdot |B(\mathbf{0}, t_e)| = q^{k_e} \sum_{i=0}^{t_e} \binom{n_e}{i} (q-1)^i.$$

As the ambient space contains q^{n_e} vectors in total, the probability that a uniformly random vector $\mathbf{x}$ falls into one of these decoding regions is

$$\text{FPR} = \frac{q^{k_e}}{q^{n_e}} \sum_{i=0}^{t_e} \binom{n_e}{i} (q-1)^i,$$

which completes the proof. $\square$

By substituting the FPRs of $\mathcal{C}_1$ and $\mathcal{C}_2$ into Eq. (6), the overall FPR of Conc-CEC is expressed as their product. We will provide the specific FPR of Conc-CEC under concrete parameters in Section 7.6, where we also provide a comparative study with other existing schemes. As demonstrated in the evaluation, the concatenated structure of Conc-CEC effectively maintains a low FPR while achieving high error-correction robustness.

7 Application: CEC-based Watermarks

Due to the favorable properties of CECs, we present their use in watermarking for AIGCs as a representative application.

7.1 CECMark

We propose CECMark, an in-processing watermarking scheme for diffusion models. Specifically, CECMark embeds watermark information in the initial latents using Conc-CEC, where an auxiliary mask mask_1 is applied to improve the quality of the generated outputs. The formal procedures are presented in Algorithm 1, Algorithm 2 and Algorithm 3. Illustrations of CECMark are provided in Appendix I.

Watermark Key Generation. Let N be the length of the initial latent and $[N] = \{i \mid 1 \leq i \leq N\}$. In Algorithm 1, the

Algorithm 1 CECMark.KeyGen

Input: Security parameter λ

Output: Master key $\mathbf{mk} \in \mathbb{F}_2^{2\lambda_1}$, permutation $\mathbf{P} : [N] \rightarrow [N]$

1: $\mathbf{mk} \leftarrow \text{Conc-CEC.KeyGen}(1^\lambda)$

2: $\mathbf{P} \xleftarrow{\$} \mathcal{P}_N$ $\triangleright \mathcal{P}_N$ is the set of all bijections over $[N]$

3: **return** $\mathbf{mk}, \mathbf{P}$

Algorithm 2 CECMark.Embed

Input: Security parameter λ , initial latent $\mathbf{z} \in \mathbb{R}^N$, message $\mathbf{m} \in \mathbb{F}_2^k$, master key $\mathbf{mk} \in \mathbb{F}_2^{2\lambda_1}$, system permutation $\mathbf{P} \in \mathcal{P}_N$

Output: Watermarked latent $\mathbf{z}' \in \mathbb{R}^N$

1: $\mathbf{c} \leftarrow \text{Conc-CEC.Encrypt}(1^\lambda, \mathbf{mk}, \mathbf{m})$ $\triangleright$ Ciphertext of length n

2: $\mathbf{s} \xleftarrow{\$} c \parallel \text{RandBits}(N-n)$ $\triangleright$ Pad to length N

3: $\mathbf{v} \leftarrow \text{Sort}(|\mathbf{z}|, \text{descending})$ $\triangleright$ Sort magnitude

4: $\mathbf{v} \xleftarrow{\$} \text{Shuffle}(\mathbf{v}_{1:n}) \parallel \text{Shuffle}(\mathbf{v}_{n+1:N})$ $\triangleright$ Shuffle significant and insignificant parts

5: $\mathbf{z}' \leftarrow (2\mathbf{s} - 1) \cdot \mathbf{v}$ $\triangleright$ Embed into signs

6: $\mathbf{z}' \leftarrow \mathbf{P}(\mathbf{z}')$ $\triangleright$ Apply the global permutation

7: **return** $\mathbf{z}'$

master key of Conc-CEC is generated in Line 1, while a uniformly random permutation $\mathbf{P}$ over $[N]$ is sampled in Line 2. The master key must remain secret to ensure security, whereas $\mathbf{P}$ is random but does not need to be kept confidential.

Watermark Embedding. In Algorithm 2, CECMark.Embed first encrypts $\mathbf{m}$ to $\mathbf{c}$ via Conc-CEC (Line 1). Given an initial latent vector $\mathbf{z}$, we sort its coordinates by absolute value and apply a permutation that maps the n largest-magnitude positions to the prefix (Line 2-4). This step follows the optimization strategy proposed in [40], which restricts embedding to coordinates that remain stable under noise perturbations. We then embed $\mathbf{c}$ by modulating the signs of these n coordinates (Line 5). Finally, a global permutation $\mathbf{P}$ is applied to decorrelate the embedding locations (Line 6). Without this step, localized perturbations in the image domain may induce burst errors in the latent space, violating the independence assumption commonly used in DFR analysis. The global permutation mitigates such structured errors and improves robustness.

Watermark Extraction. As shown in Algorithm 3, given a recovered latent vector, CECMark.Extract($\cdot$) first applies the inverse permutation $\mathbf{P}^{-1}$ to restore the canonical ordering (Line 1). The watermark bits are then directly recovered from the signs of the first n coordinates in Line 2. In Line 3, the resulting vector is decrypted by Conc-CEC, yielding either the embedded message or the failure symbol $\perp$.

Algorithm 3 CECMark.Extract

Input: Security parameter λ , recovered latent $\hat{\mathbf{z}} \in \mathbb{R}^N$, master key $\mathbf{mk} \in \mathbb{F}_2^{2\lambda_1}$, system permutation $\mathbf{P} \in \mathcal{P}_N$
Output: Extracted message vector $\hat{\mathbf{m}} \in \mathbb{F}_2^k$ or $\perp$
1: $\hat{\mathbf{z}} \leftarrow \mathbf{P}^{-1}(\hat{\mathbf{z}})$ $\triangleright$ Reverse the global permutation
2: $\hat{\mathbf{c}} \leftarrow \text{Sign}(\hat{\mathbf{z}}_{1:n})/2 + 0.5$ $\triangleright$ Extract ciphertext
3: $\hat{\mathbf{m}} \leftarrow \text{Conc-CEC.Decrypt}(1^\lambda, \mathbf{mk}, \hat{\mathbf{c}})$ $\triangleright$ Message or $\perp$
4: **return** $\hat{\mathbf{m}}$

Table 2: Parameters of CECMark Variants.

	CECMark-0	CECMark-1	CECMark-2
n_1	4,096	2,560	2,048
n_2	12,288	9,728	6,144
RS_1	[16, 8, 9]	[10, 8, 3]	[16, 8, 9]
RM_1	[256, 8, 128]	[256, 8, 128]	[128, 8, 64]
RS_2	[48, 32, 17]	[38, 32, 7]	[48, 32, 17]
RM_2	[256, 8, 128]	[256, 8, 128]	[128, 8, 64]
Occupancy	100%	75%	50%

7.2 Experimental Setups

Text-to-image Model. We use Stable-Diffusion-2.1-Base [33] to implement watermarking schemes. The size of the initial latent is $n = 4 \times 64 \times 64$ and the output image size is 512×512 . For image generation, we set generation steps to $T = 50$ with DPMSolver [27] under a guidance scale of $\gamma = 3.0$. For latent inversion, we adopt the DPMinversion-Solver [23] for $T' = 50$ steps under a null prompt condition. We sample 1,000 prompts from the test split of the Stable Diffusion Prompts Dataset [20] to generate images.

Parameters of CECMark. We set the security parameters as $k = \lambda_1 = 256$ and $\lambda_2 = 64$. This configuration aligns with NIST Category 5 to ensure the highest post-quantum security level while providing sufficient capacity for practical payloads, with the 2^{64} randomness space being more than sufficient throughout the watermark’s lifecycle. This is consistent with standard practice in symmetric encryption constructions built from block ciphers such as AES [12], where initialization vectors or nonces are typically much shorter than the underlying security parameter. We instantiate the pseudo-random functions f_1 and f_2 using ChaCha20’s keystream generator [5], which is parameterized by a 256-bit key and a 96-bit nonce, i.e., we set

$$f_1 = \text{ChaCha20.KeyStreamGenerator}(\mathbf{mk}_1, 1^{96}),$$

$$f_2 = \text{ChaCha20.KeyStreamGenerator}(\mathbf{mk}_2, 0^{32} \parallel \theta).$$

To evaluate the impact of embedding capacity and robustness, we further instantiate three concrete watermark configurations with different channel lengths, as summarized in Table 2.

Efficiency of CECMark. CECMark achieves 243–250 it/s (iterations per second) throughput for encoding and 7.79–10.46 it/s for decoding, as RM-RS decoding is more complex

than encoding. This asymmetry is acceptable since encoding is much more frequent in practice. Moreover, compared to image generation (0.56 it/s) and latent inversion (0.85 it/s), the overhead of CECMark is negligible. The detailed analysis is in Appendix H.

Baseline Watermarking Schemes. We adopt three high-capacity in-processing watermarking schemes as baselines: GS [41], PRCMark [19], and T2SMark [40]. Since the performance of watermark schemes is closely tied to their encoding capacity, we fix the payload size to 256 bits across all schemes to ensure a fair comparison. All payloads are mutually distinct across generations. Detailed settings of each scheme are in Appendix F.

7.3 Fidelity of CECMark

Image Quality Metrics. We quantitatively evaluate image quality using Fréchet Inception Distance (FID) [22] and CLIP score [21]. FID measures the distributional discrepancy between reference and evaluated images, with lower values indicating closer similarity. We fix the reference set to 1,000 images generated by the original pipeline. CLIP score measures the semantic alignment between the generated images and their prompts, where higher values reflect stronger fidelity. We employ a 1.3B CLIP model, CLIP-ViT-g-14-laion2B-s12B-b42K [34], to evaluate similarity for each image-prompt pair and compute the average across all pairs.

Consistency Metrics. To assess the consistency between unwatermarked and watermarked images, we compute FID and CLIP scores for the original pipeline and all watermarking schemes over 10 trials, each with a fresh seed or keypair. We then perform two-sample t-tests with hypotheses $H_0 : \mu_1 = \mu_0$ and $H_1 : \mu_1 \neq \mu_0$, where μ_0 and μ_1 denote the mean metric values for unwatermarked and watermarked images. A higher p -value provides stronger support for H_0 .

Evaluation Results. The evaluation results are shown in Table 3, and the images generated by different schemes are shown in Figure 13. For FID, all watermark schemes yield p -values below 0.05, indicating that watermarking alters the distributional similarity compared to the reference watermarked images. However, CECMark achieves the highest p -value among all schemes, suggesting greater consistency with unwatermarked images. For the CLIP score, all schemes show no statistically significant difference from the baseline except GS. Notably, CECMark-0 demonstrates both the highest semantic fidelity and the strongest consistency with unwatermarked images. Overall, CECMark preserves semantic alignment while exerting a smaller impact on distributional similarity compared to other watermarking schemes.

7.4 Confidentiality of CECMark

Theoretical Analysis. The confidentiality of different watermarking schemes is summarized in Table 4. (1) In GS, there

Table 3: Quality of watermarked images. We report the mean and standard error (SE) of each metric, together with the p -value between watermarked and non-watermarked images.

Schemes	FID $\downarrow$ (p -value $\uparrow$)	CLIP Score $\uparrow$ (p -value $\uparrow$)
w/o. WM	50.413 ± 0.088 (—)	34.637 ± 0.026 (—)
GS	53.471 ± 0.244 (0.000)	34.521 ± 0.048 (0.047)
PRCMark	50.853 ± 0.093 (0.003)	34.593 ± 0.038 (0.359)
T2SMark	51.269 ± 0.096 (0.000)	34.586 ± 0.042 (0.321)
CECMark-0	50.965 ± 0.107 (0.001)	34.617 ± 0.023 (0.576)
CECMark-1	<u>50.866 ± 0.145</u> (0.015)	34.541 ± 0.045 (0.085)
CECMark-2	50.986 ± 0.146 (0.003)	34.578 ± 0.020 (0.091)

exists no randomness¹, i.e., when the message is fixed, the signs of the initial latent are fixed. In the IND-CPA game, we can simply query the oracle to encrypt $\mathbf{m}_0$ again and obtain the ciphertext $\mathbf{c}_0$. If $\mathbf{c}^* = \mathbf{c}_0$, output $b = 0$. Otherwise, output $b = 1$. The success probability is 100% with $O(1)$ queries, i.e., $O(n)$ complexity. (2) As discussed in [39], PRCMark suffers from a $\binom{n}{t/2}$ key recovery attack. By modifying the key recovery attack, the adversary can obtain a partial secret key and further decode the message with accuracy comparable to that of the secret-key holder. The details of the attack are provided in Appendix M. (3) Since the length of the master key $\mathbf{mk}_{T2S}$ of T2SMark is only 16 bits, an adversary can directly enumerate the master key and recover the session key to break the scheme. The adversary could also directly enumerate the 16-bit session key $\mathbf{sk}_{T2S}$ to recover the message. Therefore, the security of T2SMark is determined by the minimum length of the master key and the session key. (4) The security of CECMark depends only on the length of the master key, while the nonce segment of the latent representation mainly serves to provide diversity and prevent reuse.

Empirical Distinguishability. To evaluate real-world confidentiality, following [19, 40], we train a ResNet-18 binary classifier for each scheme to distinguish watermarked images from unwatermarked images. The results show that CECMark-0 achieves the lowest distinguishability among all watermark schemes. The detailed settings and results are provided in Appendix K.

7.5 Robustness of CECMark

Evaluation Metric. We evaluate the robustness of watermarking schemes using the True Positive Rate (TPR), defined as the proportion of watermarked images that are still correctly identified as watermarked after undergoing various perturbations. Note that the verification is evaluated in the multi-bit

¹ Although [38] discusses some GS variants that could achieve IND-CPA security, they require recording all previously used nonces and performing decryption by enumerating them, which is highly impractical in realistic settings.

Table 4: Comparison of theoretical and concrete time complexities against confidentiality across different schemes.

Scheme	Theoretical Time	Concrete Time
GS	$O(n)$	2^{14}
PRCMark	$O(\binom{n}{t/2})$	2^{28}
T2SMark	$O(2^{\min(\mathbf{mk}_{T2S} , \mathbf{sk}_{T2S})})$	2^{16}
CECMark	$O(2^{\lambda_1})$	2^{256}

Table 5: Robustness of watermarking schemes measured by True Positive Rate (TPR).

Schemes	TPR@Clean $\uparrow$	TPR@Perturbed $\uparrow$
GS	0.9689 ± 0.0431	0.4959
PRCMark	0.9994 ± 0.0007	0.7277
T2SMark	0.9863 ± 0.0047	0.5197
CECMark-0	<u>0.9998 ± 0.0004</u>	0.8027
CECMark-1	1.0000 ± 0.0000	0.8339
CECMark-2	1.0000 ± 0.0000	<u>0.8148</u>

setting, where a true positive is counted only if the decoded message perfectly equals the original embedded message.

Evaluation Results. As a baseline, we first evaluate the clean TPR on 1,000 unperturbed watermarked images. This evaluation is repeated across 10 trials, each with a fresh keypair. We then evaluate robustness under 56 perturbations spanning diverse transformations and strength levels. Detailed settings are explained in Appendix G. The overall clean and perturbed TPRs are reported in Table 5 and the per-perturbation results are provided in Figure 8. Evaluation results demonstrate that CECMark consistently achieves higher robustness than existing watermark schemes. Among the three parameter sets, CECMark-1 achieves the highest robustness, while CECMark-0 performs the lowest, highlighting the effectiveness of the selective embedding strategy.

Decoding Boundary Analysis. To characterize the decoding boundaries of different schemes, we employ the ratio of flipped signs between original and inverted latents as a unified measure of perturbation strength. We report the corresponding TPRs under varying sign-flip ratios in Figure 4. The results show that CECMark maintains a stable TPR until the sign-flip ratio reaches 37%, and its TPR drops to zero around 40%. In contrast, the strongest baseline PRCMark begins to degrade at 33% and drops to zero near 39%. Consequently, CECMark achieves a decoding boundary approximately 10% larger than the existing schemes, thereby achieving better watermark robustness.

7.6 Soundness of CECMark

For the baseline watermarking schemes, the soundness is either not achieved or costly. We summarize the soundness

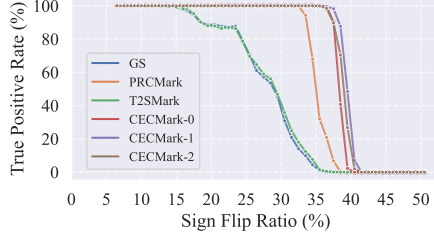


Figure 4: True Positive Rate (TPR) of watermarks under different latent sign flip ratios.

of different schemes in Table 6. Empirically, we observe zero false positives across 10,000 unwatermarked images for all watermarking schemes. However, CECMark achieves a low theoretical FPR without introducing any other side effects.

Soundness of GS. GS works in either detection mode or decoding mode. In detection mode, GS encodes a fixed message and detects whether the watermark exists with a threshold. In decoding mode, even if the input image were not watermarked, the decoding result would be a valid message. Therefore, soundness is achieved only in the detection version of GS.

Soundness of PRCMark. PRCMark chooses extra parity bits to bound the FPR, i.e., after decoding, the result is accepted if and only if the parity check bits are identical to the preset value. While this approach guarantees a fixed FPR, it inherently increases the code rate. In Belief Propagation (BP)-based decoders, a higher code rate typically reduces the error-correction capability, thereby undermining the overall robustness of the scheme.

Soundness of T2SMark. Because the decoding method for duplicated codes could be applied to any codeword, T2SMark has to calculate a confidence level to detect the watermark. A high threshold weakens robustness by making the watermark easier to remove, whereas a low threshold increases false positives on unwatermarked images. T2SMark therefore faces an inherent trade-off between robustness and soundness, which can be regarded as a side effect.

Table 6: Comparison of soundness, side effect, and target FPR across different watermarking schemes.

Scheme	Soundness	Side Effect	Target FPR
GS	● (Detection only)	-	-
PRCMark	●	Reduced robustness	10^{-5}
T2SMark	●	Reduced robustness	10^{-6}
CECMark-0	●	None	10^{-17}
CECMark-1	●	None	10^{-5}
CECMark-2	●	None	10^{-17}

8 Discussions

8.1 Comparison with Duplicated Codes

As duplicated codes are applied in GS and T2SMark, we carry out a theoretical analysis of the DFR for duplicated codes.

Theorem 5 (DFR of Duplicated Codes). *Let n denote the code length, k the message length, and $d = n/k$ the duplication factor. Let p be the noise rate of the noisy channel. The DFR of the duplicated code is as follows.*

If d is odd, the DFR is

$$1 - \left[1 - \sum_{i=\lfloor d/2 \rfloor + 1}^d \binom{d}{i} p^i (1-p)^{d-i} \right]^k.$$

If d is even, the DFR is

$$1 - \left[1 - \sum_{i=d/2+1}^d \binom{d}{i} p^i (1-p)^{d-i} - \frac{1}{2} \binom{d}{d/2} p^{d/2} (1-p)^{d/2} \right]^k.$$

Proof. Each message bit is repeated $d = n/k$ times. When d is odd, decoding fails if and only if more than half of its copies are flipped. The corresponding probability is

$$p_0 = \sum_{i=\lfloor d/2 \rfloor + 1}^d \binom{d}{i} p^i (1-p)^{d-i}.$$

When d is even, in addition to the above case, a decoding failure also occurs with $1/2$ probability if exactly half of the positions are flipped. The corresponding probability is

$$p_0 = \sum_{i=\lfloor d/2 \rfloor + 1}^d \binom{d}{i} p^i (1-p)^{d-i} + \frac{1}{2} \binom{d}{d/2} p^{d/2} (1-p)^{d/2}.$$

A decoding failure occurs if and only if at least one of the duplicated codes is decoded incorrectly. Thus,

$$\text{DFR} = 1 - (1 - p_0)^k,$$

which completes the proof. $\square$

DFR Comparison Results. We then compare the DFR of the RM-RS code used in our CECMark scheme with that of the duplicated code adopted in GS and T2SMark. For a fixed message length and code length, we vary the channel error rate and evaluate the resulting DFR over 10^4 trials for each setting. Specifically, we consider the message length $k = 256$ and the code length $n \in \{6144, 9728, 12288\}$, corresponding to the parameters k, n_2 in CECMark-2, CECMark-1, and CECMark-0, respectively. As shown in Figure 5, for the duplicated code, the theoretical DFR matches the empirical results exactly. For the RM-RS code, the theoretical DFR serves as an upper bound across all parameters, indicating that the analysis is conservative and the actual performance is better than predicted. Across all tested error rates, the RM-RS code consistently achieves a lower DFR than the duplicated code, both theoretically and empirically.

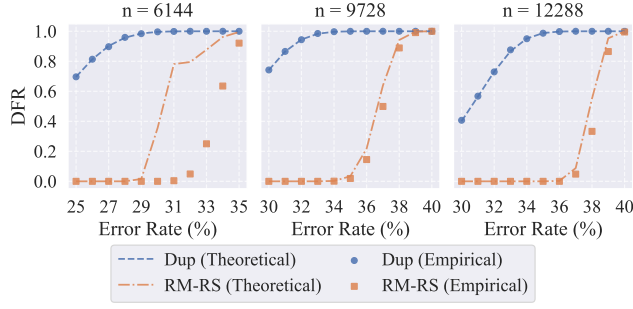


Figure 5: DFR comparison between duplicated codes and RM-RS codes with different n .

8.2 Comparison with PRCs

Pseudorandom error-correcting Codes (PRCs) [9] is the most closely related cryptographic primitive to CECs and is also a promising primitive for AIGC watermarking. We therefore provide a dedicated comparison with PRCs to highlight the advantages of CECs and explain why CECs are needed as a distinct primitive.

Property-Level Comparison. PRCs [9] are required to simultaneously satisfy three properties: undetectability, robustness, and soundness. These properties closely align with the requirements of CECs, except that PRCs rely on undetectability whereas CECs rely on confidentiality. In the context of watermarking, undetectability serves two roles: it helps preserve the perceptual quality of generated content and also provides a form of information hiding. However, undetectability is a strong requirement, and existing PRC instantiations have been shown to lose this property under practical parameters [39]. Moreover, as indicated in Section 7.5, enforcing undetectability may impose additional constraints on robustness. In contrast, CECs formalize information protection directly through confidentiality. Meanwhile, Section 7.3 shows that CECMark can also preserve high perceptual quality in practice.

Instantiability-Level Comparison. As far as we know, although the works in [8, 9, 16, 19, 31] claim that they can successfully construct PRCs, all of these constructions have flaws in choosing concrete parameters or fail to balance robustness and undetectability. In Appendix L, we will conduct a detailed analysis of their limitations. Overall, while PRCs are theoretically attractive, their practical performance is still limited.

Building PRCs based on CECs. In Conc-CEC, the RM-RS code is inherently distinguishable from random due to its strong algebraic structure, and thus does not provide undetectability. In our framework, undetectability depends on whether the underlying code is computationally indistinguishable from random. While one may replace RM-RS with pseudorandom codes such as LDPC or MDPC combined with noise, prior attacks [39] indicate that achieving both strong undetectability and robustness in practice remains challenging.

Therefore, achieving full PRC security with suitable codes remains an interesting direction for future work.

8.3 Other Possible Applications

Following [9, 19], we regard watermarking as our main application scenario. The primary challenge in applying CECs to Large Language Model (LLM) watermarking is how to embed the Conc-CEC codewords into token sequences. A common approach adopted by PRCs [9] is to let each watermark bit influence the sampling procedure of individual tokens. However, as analyzed in [39], under reasonable text quality constraints, the entropy of output tokens is too low, resulting in a noise rate that remains an unsolved challenge for existing error-correction decoders. Moreover, constructing an edit-robust watermark for LLMs often incurs prohibitive overhead. Specifically, existing schemes like [9] employ majority encoding for each watermark bit to resist the deletion channel, which substantially expands the watermark length and renders the construction impractical for deployment in real-world LLMs. While Conc-CEC could theoretically be extended to resist deletion and insertion errors via similar majority encoding techniques, we currently restrict its application to GIMs due to these practical constraints. We leave the concrete optimization and implementation of CECs for LLM watermarking as future work. We believe that CECs may also be applicable to more applications such as watermarking video diffusion models [24], which represents a promising direction for future research.

9 Conclusion

We present a new primitive, Confidential Error-Correcting Codes (CECs), a symmetric-key primitive that integrates encryption with error-correcting capability. The confidentiality, robustness, and soundness of CECs are designed to address the key challenges in watermarking generative models. We further develop a CEC construction framework based on a randomness-security separation structure, reducing confidentiality to pseudorandom functions, while ensuring robustness and soundness with error-correcting codes. Instantiated with concatenated codes, we present Conc-CEC and analyze its security properties theoretically. Finally, we design CECMark, a watermarking scheme for generative image models based on Conc-CEC. Experiments show that CECMark maintains strong robustness and soundness and achieves consistently higher detection accuracy than existing methods under various image perturbations. Looking forward, we believe that CECs can serve as a foundational building block for trustworthy watermarking systems, enabling the field to move from heuristic designs toward cryptographically grounded and practically reliable solutions.

References

- [1] Adil S. Al-Busaidi, Raghu Raman, Laurie Hughes, Mousa Ahmed Albashrawi, Tegwen Malik, Yogesh K. Dwivedi, Thuraiya Al- Alawi, Mohammed AlRizeiqi, Gareth Davies, Mark Fenwick, Parul Gupta, Shashikala Gurpur, Apeksha Hooda, Paulius Jurcys, Daryl Lim, Nicola Lucchi, Tanvi Misra, Ramakrishnan Raman, Anuragini Shirish, and Paul Walton. Redefining boundaries in innovation and knowledge domains: Investigating the impact of generative artificial intelligence on copyright and intellectual property rights. *Journal of Innovation & Knowledge*, 9(4):100630, 2024.
- [2] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: Benchmarking the robustness of image watermarks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 1456–1492. PMLR, 21–27 Jul 2024. URL: <https://proceedings.mlr.press/v235/an24a.html>.
- [3] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, et al. Bike: bit flipping key encapsulation. 2022.
- [4] Elaine B Barker, John Michael Kelsey, et al. *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2007.
- [5] Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pages 3–5. Lausanne, Switzerland, 2008.
- [6] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 79–158. Springer, 2017.
- [7] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European symposium on security and privacy (EuroS&P)*, pages 353–367. IEEE, 2018.
- [8] Miranda Christ, Noah Golowich, Sam Gunn, Ankur Moitra, and Daniel Wichs. Improved pseudorandom codes from permuted puzzles. *arXiv preprint arXiv:2512.08918*, 2025.
- [9] Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024*, volume 14925 of *Lecture Notes in Computer Science*, pages 325–347, Santa Barbara, CA, USA, 2024. Springer.
- [10] Claude. How claude marks ai-generated content. URL: <https://support.claude.com/>.
- [11] Coalition for Content Provenance and Authenticity. C2pa. URL: <https://c2pa.org/>.
- [12] Joan Daemen and Vincent Rijmen. *The design of Rijndael*, volume 2. Springer, 2002.
- [13] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act), 2024.
- [14] Danilo Francati, Yevin Nikhel Goonatilake, Shubham Pawar, Daniele Venturi, and Giuseppe Ateniese. The coding limits of robust watermarking for generative models. *Cryptology ePrint Archive*, Paper 2025/1620, 2025.
- [15] Philippe Gaborit, Carlos Aguilar-Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Edoardo Persichetti, Gilles Zémor, Jurjen Bos, et al. Hamming quasi-cyclic (hq). *NIST Post-Quantum Standardization, 4th Round; National Institute of Standards and Technology: Gaithersburg, MD, USA*, 2025.
- [16] Surendra Ghentiyala and Venkatesan Guruswami. New Constructions of Pseudorandom Codes. In Alina Ene and Eshan Chattopadhyay, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, volume 353 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:22, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2025.54>, doi:10.4230/LIPIcs.APPROX/RANDOM.2025.54.
- [17] Google DeepMind. Synthid: Identifying ai-generated content. URL: <https://deepmind.google/technologies/synthid/>.

- [18] Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, Chris Gamble, Miklós Z. Horváth, Fabian Kaczmarczyk, Alex Kaskasoli, Aleksandar Petrov, Ilia Shumailov, Meghana Thotakuri, Olivia Wiles, Jessica Yung, Zahra Ahmed, Victor Martin, Simon Rosen, Christopher Savčák, Armin Senoner, Nidhi Vyas, and Pushmeet Kohli. Synthid-image: Image watermarking at internet scale, 2025. [arXiv:2510.09263](https://arxiv.org/abs/2510.09263).
- [19] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- [20] Gustavosta. Stable-Diffusion-Prompts, 2022. URL: <https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts>.
- [21] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.emnlp-main.595.
- [22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [23] Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hye-won Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7069–7078, 2024.
- [24] Xuming Hu, Hanqian Li, Jungang Li, Yu Huang, Shuliang Liu, Qi Zheng, Junhao Chen, and Aiwei Liu. Videomark: A distortion-free robust watermarking framework for video diffusion models. *arXiv preprint arXiv:2504.16359*, 2025.
- [25] Poonam Jindal and Brahmjit Singh. Rc4 encryption-a literature survey. *Procedia Computer Science*, 46:697–705, 2015.
- [26] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: Keyed-hashing for message authentication. Technical report, 1997.
- [27] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, Aug 2025.
- [28] Nicola Lucchi. Chatgpt: A case study on copyright challenges for generative artificial intelligence systems. *European Journal of Risk Regulation*, 15(3):602–624, 2024.
- [29] Chetan Nanjunda Mathur, Karthik Narayan, and KP Subbalakshmi. On the design of error-correcting ciphers. *EURASIP Journal on Wireless Communications and Networking*, 2006(1):042871, 2007.
- [30] Microsoft. Content credentials in azure openai. URL: <https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/content-credentials>.
- [31] Ankur Moitra and Noah Golowich. Edit distance robust watermarks for language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 20645–20693, 2024.
- [32] OpenAI. C2pa and synthid in openai-generated images. URL: <https://help.openai.com/en/articles/8912793-c2pa-in-images>.
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 6 2022.
- [34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pages 25278–25294, 2022.
- [35] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramanian, John Paul Strachan, Miao Hu, R Stanley Williams, and Vivek Srikumar. Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. *ACM SIGARCH Computer Architecture News*, 44(3):14–26, 2016.

- [36] ShieldMnt. invisible-watermark: python library for invisible image watermark, 2021.
- [37] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegas-tamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [38] Jonas Thietke, Andreas Müller, Denis Lukovnikov, Asja Fischer, and Erwin Quiring. Towards a correct usage of cryptography in semantic watermarks for diffusion models. In *The 1st Workshop on GenAI Watermarking*, 2025. URL: <https://openreview.net/forum?id=yENK6DYuXv>.
- [39] Tianrui Wang, Anyu Wang, Tianshuo Cong, Delong Ran, Jinyuan Liu, and Xiaoyun Wang. Cryptanalysis of ldpc-based pseudorandom error-correcting codes. In *35th USENIX Security Symposium (USENIX Security 26)*. USENIX Association, August 2026.
- [40] Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. T2SMARK: Balancing robustness and diversity in noise-as-watermark for diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [41] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024.
- [42] Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairuze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, Somesh Jha, Lei Li, Yu-Xiang Wang, and Dawn Song. SoK: Watermarking for AI-Generated Content. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 2621–2639, 2025.

A Open Science

All authors have read and consented to the open science policy. We are committed to releasing our artifacts to the community to ensure reproducibility and to benefit the broader research ecosystem. The artifacts are anonymously available at: <https://anonymous.4open.science/r/USENIX-CECMARK-6E87/>.

Code Availability. The artifact includes implementations for:

- CECMark and the underlying Conc-CEC scheme;
- Baseline watermark schemes, including GS, PRCMark, and T2SMARK;

- Evaluation metrics, such as CLIP, FID, decoding correctness, latent sign flipping, and distinguishability;
- Statistical analysis and figure generation.

All code is clearly documented in the README file, and the project structure is carefully organized to ensure clarity, reproducibility, and extensibility for future research.

Data Availability. Due to the large size of the generated datasets, we do not include the full set of images and corresponding latents in the artifact. Instead, we provide a one-request demo dataset for each scheme. We also provide the statistical results over the full dataset to reproduce the figures in the paper. Public resources, such as the Stable Diffusion Prompt dataset and the Stable Diffusion 2.1 Base model, are excluded from the artifact.

B Ethical Considerations

This work introduces a new cryptographic primitive named CEC, and applies it to design CECMark for watermarking AI-generated images of generative image models. While CECMark improves the confidentiality, robustness and soundness of the watermark, several ethical considerations remain to be discussed.

Privacy Risks of Embedded Information. The watermarking techniques might be used to embed information such as user identifiers or provenance data, which introduces potential privacy risks if used without proper authorization. Although CECMark provides confidentiality against adversaries without access to the secret key, these risks stem from how watermarking is deployed rather than the embedding itself. To mitigate these concerns, users should be clearly informed and give explicit consent before their information is embedded into watermarks, and such mechanisms should only be used for legitimate purposes such as provenance or attribution.

Security of Pseudorandom Function Instantiation. As shown in our work, the confidentiality of CECMark is reduced to the security of pseudorandom functions. When deploying watermarking systems in practice, we suggest that cryptographically secure pseudorandom functions should be used. If model providers replace pseudorandom functions with non-cryptographic random generators for simplicity, it may violate the underlying security assumptions and lead to potential information leakage.

Impact Control of Theoretical Attacks. Although we reveal the limitations of existing watermarking schemes including GS, PRCMark, and T2SMARK, our goal is to avoid the deployment of schemes with potential security risks. In our discussions, the confidentiality of these schemes is not satisfied under practical parameter sets. Since these watermarking schemes are not yet deployed in real-world environments, our theoretical attacks did not cause any real-world privacy leaks or other negative effects. As a result, we propose CEC

with provable confidentiality to prevent potential attacks and support the secure deployment of watermarking schemes.

Responsible Deployment and Usage. We emphasize that watermarking should only be applied in appropriate application scenarios with clearly defined purposes. System providers should carefully evaluate the impact of watermarking on system behavior and ensure that its deployment aligns with the intended functionality of generative models. More broadly, watermarking is not a complete solution to the misuse of generative models, and should be combined with other mechanisms for content governance and authenticity.

C Related Works

C.1 Error-Tolerant Cryptography

The cryptographic schemes with error-tolerant capability are mainly divided into two categories. (1) **Public-key cryptography** based on codes and lattices inherently involves noise during encryption, such as BIKE [3], HQC [15], and Kyber [7]. However, the injected noise in these schemes is primarily used to ensure security, rather than to provide robustness against adversarial perturbations. Moreover, the magnitude and distribution of the noise are tightly coupled with the underlying algebraic structures of codes or lattices, and must be carefully controlled to guarantee correct decryption. As a result, the tolerable error rate is inherently limited. For example, the error rates in BIKE and HQC are around 0.5% and 0.4%, respectively, whereas the error rate under watermark removal attacks is on the order of 10%, which is significantly larger. Therefore, such schemes offer limited robustness in applications like watermarking. (2) **Secret-key cryptography** can also be combined with error-correcting codes. [29] combines AES encryption with Reed-Solomon codes to achieve secure and reliable communication over wireless channels. However, their construction mainly focuses on integrating error-correcting capability with a deterministic block cipher, without explicitly considering the randomized encryption mechanisms required for IND-CPA security. In particular, modern symmetric encryption schemes typically rely on a fresh nonce or initial vector for each encryption, which is treated as part of the ciphertext. Such randomness is not incorporated into the design or analysis in [29]. Therefore, their construction does not capture the setting considered in our work, where both randomness and error tolerance must be jointly handled within the encryption process.

C.2 Post-processing Watermarks

Post-processing schemes embed watermarks into images after they are generated by GIMs [18, 36, 37]. Such schemes are generally implemented using one of two approaches: steganography-based or encoder-decoder-based [2]. Dwt-Dct [36] is one of the most widely used steganography-based

methods, which combines the Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) to encode the message into the frequency domain of the target image. However, this technique often introduces visible artifacts and lacks provable robustness guarantees. The encoder-decoder-based approach trains an encoder for watermark embedding and a decoder for watermark detection [18, 37]. A representative example is SynthID-Image [18], which is a production-level private watermark scheme developed by Google DeepMind. While this paradigm can achieve a practical balance between image quality and watermark robustness, it lacks provable security and soundness guarantees. Consequently, it remains susceptible to black-box watermark removal attacks².

D Encoding Procedure of Concatenated Codes

The encoding procedure $\mathbf{v} = \mathcal{C}.\text{Encode}(\mathbf{m})$ of the concatenated code is shown in Figure 6 and the decoding procedure follows analogously.

Let $q = 2^{k_i}$ and $f : \mathbb{F}_q \rightarrow \mathbb{F}_2^{k_i}$ be a fixed bijection that maps each element in $\mathbb{F}_q$ to its k_i -bit representation over $\mathbb{F}_2$. The message space of the concatenated code is $\mathbb{F}_2^{k_e k_i}$. Given a binary message $\mathbf{m} \in \mathbb{F}_2^{k_e k_i}$, we partition it into k_e blocks $\mathbf{m} = (\mathbf{m}_1, \dots, \mathbf{m}_{k_e})$, where $\mathbf{m}_j \in \mathbb{F}_2^{k_i}$. Each block is mapped to a symbol in $\mathbb{F}_q$ via f^{-1} , yielding

$$\mathbf{m}^{(q)} = (f^{-1}(\mathbf{m}_1), \dots, f^{-1}(\mathbf{m}_{k_e})) \in \mathbb{F}_q^{k_e}.$$

The external code encodes $\mathbf{m}^{(q)}$ as

$$\mathbf{c}^{(q)} = \mathcal{C}_e.\text{Encode}(\mathbf{m}^{(q)}) = (c_1^{(q)}, \dots, c_{n_e}^{(q)}), \quad c_j^{(q)} \in \mathbb{F}_q.$$

Next, each symbol $c_j^{(q)}$ is first mapped to $\mathbb{F}_2^{k_i}$ via f to obtain

$$\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_{n_e}), \quad \mathbf{c}_j = f(c_j^{(q)}) \in \mathbb{F}_2^{k_i},$$

and then encoded independently using the internal code,

$$\mathbf{v}_j = \mathcal{C}_i.\text{Encode}(\mathbf{c}_j) \in \mathbb{F}_2^{n_i}.$$

Finally, the overall binary codeword is obtained by concatenation:

$$\mathbf{v} = \mathcal{C}.\text{Encode}(\mathbf{m}) = (\mathbf{v}_1, \dots, \mathbf{v}_{n_e}) \in \mathbb{F}_2^{n_e n_i}.$$

E Games for the IND-CPA Proof

This section presents the modified games used in the proof of Theorem 1.

²<https://synthidbypass.com/>.

Game 4 $\text{Game}_{f_1, \mathcal{B}_1}^{\text{PRF}}(\lambda)$ (Distinguish f_1 from random)

1. $b \xleftarrow{\$} \{0, 1\}$
2. $\mathcal{O}_1(\cdot) := \begin{cases} f_1(\mathbf{mk}_1) & \text{if } b = 0 \\ R_1(\cdot) & \text{if } b = 1 \end{cases}$
3. $\mathbf{mk}_2 \xleftarrow{\$} \mathbb{F}_2^{\lambda_1}$
4. $\mathcal{O}(\mathbf{m})$:
 - (a) $\theta \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
 - (b) $\mathbf{c}_1 \leftarrow \text{Encode}_1(\theta) + \mathcal{O}_1()$
 - (c) $\mathbf{c}_2 \leftarrow \text{Encode}_2(\mathbf{m}) + f_2(\mathbf{mk}_2, \theta)$
 - (d) return $(\mathbf{c}_1, \mathbf{c}_2)$
5. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$
6. $\beta \xleftarrow{\$} \{0, 1\}$
7. $\theta^* \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
8. $\mathbf{c}_1^* \leftarrow \text{Encode}_1(\theta^*) + \mathcal{O}_1()$
9. $\mathbf{c}_2^* \leftarrow \text{Encode}_2(\mathbf{m}_\beta) + f_2(\mathbf{mk}_2, \theta^*)$
10. $\hat{\beta} \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, (\mathbf{c}_1^*, \mathbf{c}_2^*))$
11. $\mathcal{B}_1$ outputs $b' \leftarrow 1 - (\hat{\beta} = \beta)$
12. return $(b' = b)$

Game 5 $\text{Game}_{f_2, \mathcal{B}_2}^{\text{PRF}}(\lambda)$ (Distinguish f_2 from random)

1. $b \xleftarrow{\$} \{0, 1\}$
2. $\mathcal{O}_2(\cdot) := \begin{cases} f_2(\mathbf{mk}_2, \cdot) & \text{if } b = 0 \\ R_2(\cdot) & \text{if } b = 1 \end{cases}$
3. $\mathbf{r}_1 \xleftarrow{\$} \mathbb{F}_2^{\eta_1}$
4. $\mathcal{D} \leftarrow \emptyset$
5. $\mathcal{O}(\mathbf{m})$:
 - (a) $\theta \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
 - (b) $\mathbf{c}_1 \leftarrow \text{Encode}_1(\theta) + \mathbf{r}_1$
 - (c) if $\theta \notin \mathcal{D}$: $\mathcal{D}[\theta] \leftarrow \mathcal{O}_2(\theta)$
 - (d) $\mathbf{s} \leftarrow \mathcal{D}[\theta]$
 - (e) $\mathbf{c}_2 \leftarrow \text{Encode}_2(\mathbf{m}) + \mathbf{s}$
 - (f) return $(\mathbf{c}_1, \mathbf{c}_2)$
6. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$
7. $\beta \xleftarrow{\$} \{0, 1\}$
8. $\theta^* \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
9. $\mathbf{c}_1^* \leftarrow \text{Encode}_1(\theta^*) + \mathbf{r}_1$
10. if $\theta^* \notin \mathcal{D}$: $\mathcal{D}[\theta^*] \leftarrow \mathcal{O}_2(\theta^*)$
11. $\mathbf{s}^* \leftarrow \mathcal{D}[\theta^*]$
12. $\mathbf{c}_2^* \leftarrow \text{Encode}_2(\mathbf{m}_\beta) + \mathbf{s}^*$
13. $\hat{\beta} \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, (\mathbf{c}_1^*, \mathbf{c}_2^*))$
14. $\mathcal{B}_2$ outputs $b' \leftarrow 1 - (\hat{\beta} = \beta)$
15. return $(b' = b)$

Game 6 $\text{Game}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ (Replace f_1 with randomness)

1. $\mathbf{mk}_2 \xleftarrow{\$} \mathbb{F}_2^{\lambda_1}$
2. $\mathbf{r}_1 \xleftarrow{\$} \mathbb{F}_2^{\eta_1}$
3. $\mathcal{O}(\mathbf{m})$:
 - (a) $\theta \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
 - (b) $\mathbf{c}_1 \leftarrow \text{Encode}_1(\theta) + \mathbf{r}_1$
 - (c) $\mathbf{c}_2 \leftarrow \text{Encode}_2(\mathbf{m}) + f_2(\mathbf{mk}_2, \theta)$
 - (d) return $(\mathbf{c}_1, \mathbf{c}_2)$
4. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$
5. $b \xleftarrow{\$} \{0, 1\}$
6. $\theta^* \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
7. $\mathbf{c}_1^* \leftarrow \text{Encode}_1(\theta^*) + \mathbf{r}_1$
8. $\mathbf{c}_2^* \leftarrow \text{Encode}_2(\mathbf{m}_b) + f_2(\mathbf{mk}_2, \theta^*)$
9. $b' \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, (\mathbf{c}_1^*, \mathbf{c}_2^*))$
10. return $(b' = b)$

Game 7 $\text{Game}_{\mathcal{A}}^{\text{IND-CPA}}(\lambda)$ (Replace f_2 with randomness)

1. $\mathbf{mk}_2 \xleftarrow{\$} \mathbb{F}_2^{\lambda_1}$
2. $\mathbf{r}_1 \xleftarrow{\$} \mathbb{F}_2^{\eta_1}$
3. $\mathcal{D} \leftarrow \emptyset$
4. $\mathcal{O}(\mathbf{m})$:
 - (a) $\theta \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
 - (b) $\mathbf{c}_1 \leftarrow \text{Encode}_1(\theta) + \mathbf{r}_1$
 - (c) if $\theta \notin \mathcal{D}$: $\mathcal{D}[\theta] \xleftarrow{\$} \mathbb{F}_2^{\eta_2}$
 - (d) $\mathbf{s} \leftarrow \mathcal{D}[\theta]$
 - (e) $\mathbf{c}_2 \leftarrow \text{Encode}_2(\mathbf{m}) + \mathbf{s}$
 - (f) return $(\mathbf{c}_1, \mathbf{c}_2)$
5. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}_{\text{CHOOSE}}^{\mathcal{O}}(1^\lambda)$
6. $b \xleftarrow{\$} \{0, 1\}$
7. $\theta^* \xleftarrow{\$} \mathbb{F}_2^{\lambda_2}$
8. $\mathbf{c}_1^* \leftarrow \text{Encode}_1(\theta^*) + \mathbf{r}_1$
9. if $\theta^* \notin \mathcal{D}$: $\mathcal{D}[\theta^*] \xleftarrow{\$} \mathbb{F}_2^{\eta_2}$
10. $\mathbf{s}^* \leftarrow \mathcal{D}[\theta^*]$
11. $\mathbf{c}_2^* \leftarrow \text{Encode}_2(\mathbf{m}_b) + \mathbf{s}^*$
12. $b' \leftarrow \mathcal{A}_{\text{GUESS}}^{\mathcal{O}}(1^\lambda, (\mathbf{c}_1^*, \mathbf{c}_2^*))$
13. return $(b' = b)$

F Details of Baseline Watermarking Schemes

For GS, the payload is encoded using a duplicated code with (1,8,8) repetitions along the channel, width, and height dimensions, respectively. We use the same-key-same-nonce setting across all generations to simulate a real-world key management scenario [38]. For PRCMark, the payload is

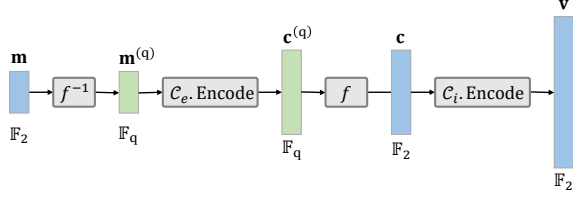


Figure 6: Encoding procedure of the concatenated codes.



Figure 7: Example images under different perturbations.

encoded using the LDPC-PRC with parameter $t = 3$ and a theoretical false positive rate of 10^{-5} . Decoding is performed using the soft-decoding technique [19] with $\sigma = 1.5$. For T2SMARK, a 16-bit session key is embedded in the first channel, while the payload is embedded across the remaining three channels. Each component is encoded with a duplicated code, ensuring 50% occupancy within its designated channels. Note that the detection and decoding mechanisms of T2SMARK are independent. We evaluate its robustness by directly decoding the message without reference to the detection result, and evaluate its soundness by empirically setting the detection threshold to 300, which produces zero false positives under incorrect keys. These settings advantage T2SMARK, yielding an overestimated upper bound in these metrics.

G Details of Watermark Robustness

Implementation of Image Transformations. We adopt 12 image transformations as shown in Figure 7, which can be roughly divided into four categories: global transformations, masking, noise, and geometric transformations. Their implementations are detailed as follows:

- **Global Transformations:**

- **JPEGCompression:** Apply JPEG compression

with quality factor α , where a higher α indicates better quality preservation.

- **Brightness:** Adjust image brightness globally. Brighter for $\alpha > 1$ and darker for $\alpha < 1$.
- **MedianFilter:** Apply a median filter with kernel size α , where a larger α results in stronger smoothing and loss of fine details.
- **GaussianBlur:** Apply Gaussian blur with standard deviation α , where a larger α produces stronger blurring effects.

- **Masking:**

- **RandCropPad:** Randomly crop a square covering $\alpha\%$ of pixels, retain the cropped region, and pad the remaining area with zeros. A larger α preserves more of the original image.
- **RandCropDrop:** Randomly crop a square covering $\alpha\%$ of pixels, and set the cropped region to zeros. A larger α drops more content from the image.

- **Noise:**

- **Salt&PepperNoise:** Randomly replace $\alpha(\%)$ of pixels with either maximum or minimum intensity values. A larger α introduces more impulsive noise.
- **GaussianNoise:** Add Gaussian noise with variance α to the image. A larger α corresponds to stronger noise levels.

- **Geometric Transformations:**

- **Rotation:** Rotate the image by α° , discarding pixels that fall outside the bounding box.
- **ScaledRotation:** Rotate the image by α° and scale it down to fit the bounding box.
- **RandCropScale:** Randomly crop a square covering $\alpha\%$ of pixels, then scale the cropped region to the full image size.
- **TranslateX:** Translate the image horizontally by $\alpha\%$ of pixels.

Per-perturbation Results. The per-perturbation TPRs of each watermark scheme are presented in Figure 8. Across all perturbations, CECMark consistently achieves higher robustness than existing watermark schemes, showing its strong robustness. Nevertheless, it still fails under certain transformations. To analyze these failures, we examine the ratios of latent sign flips induced by different perturbations in Figure 9. The results indicate that these perturbations generate nearly random latent flips, which exceed the decoding boundary of CECMark. In fact, such strong perturbations invalidate nearly any error-correcting scheme [14], representing an inherent limitation of latent-based watermarking that CECMark cannot fully resolve.

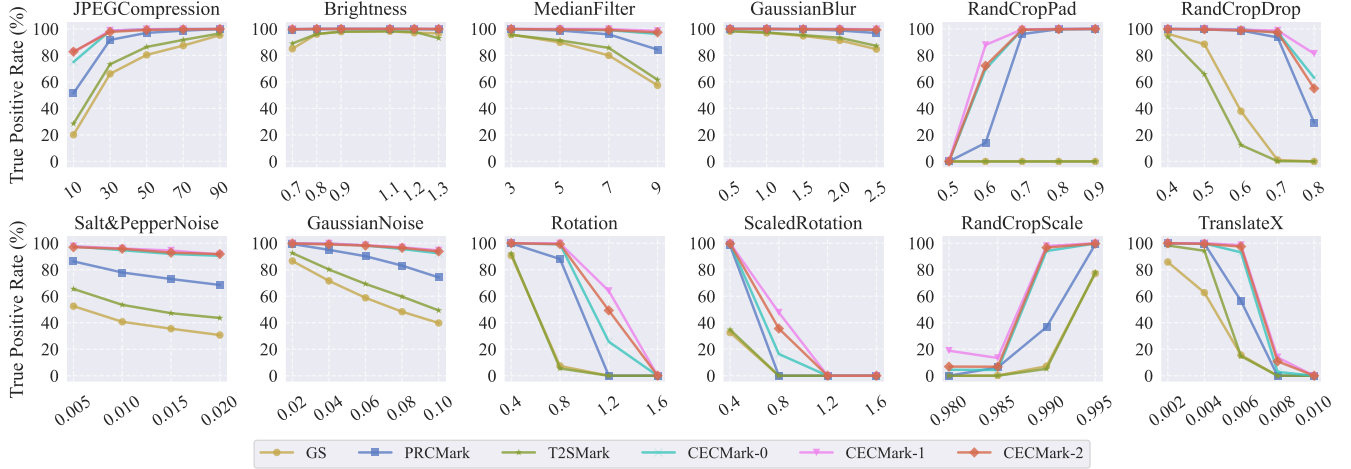


Figure 8: The True Positive Rate (TPR) of various watermark schemes under transformations of different strengths.

Table 7: The throughput of different watermark schemes for watermark encoding and decoding (iterations per second, it/s).

Schemes	Encode ($\uparrow$)	Decode ($\uparrow$)
GS	254.85	998.14
PRCMark	3.40	3.58
T2SMark	483.32	839.08
RSSMark-0	250.16	7.79
RSSMark-1	243.37	10.46
RSSMark-2	247.02	10.18

H Efficiency of Different Watermarking Schemes

The encoding and decoding throughputs (iterations per second, it/s) of different watermarking schemes are reported in Table 7. The evaluation is conducted by performing 1,000 encoding and decoding operations implemented in Python on a single process of an Intel Xeon Platinum 8558P processor.

For encoding, RSSMark achieves 243–250 it/s, with only marginal differences across variants. The throughput of RSSMark is comparable to GS, which only requires ChaCha20 bitstream generation. Although RSSMark additionally performs RM-RS encoding, this operation is linear-time and only introduces negligible overhead. T2SMark achieves the highest throughput because it uses Python’s built-in pseudo-random number generator instead of the cryptographically secure ChaCha20 stream. In contrast, PRCMark achieves only 3.40 it/s due to its expensive matrix operations.

For decoding, RSSMark achieves a low throughput of 7.79–10.46 it/s. This is because RSSMark performs computationally intensive RM-RS decoding. By comparison, GS and T2SMark employ much simpler decoding procedures and therefore achieve significantly higher throughput. Neverthe-

less, RSSMark still outperforms PRCMark, whose Belief Propagation decoder incurs even higher computational complexity.

Although CECMark exhibits asymmetric encoding and decoding throughput, this is practical because watermark encoding is performed far more frequently than watermark decoding in real-world deployments. Moreover, compared with image generation and latent inversion, which achieve only 0.56 it/s and 0.85 it/s on a single NVIDIA H20 GPU, respectively, watermark encoding and decoding are not the bottleneck of the overall pipeline.

I Illustration of Watermark Embedding and Extraction

The overall process from the initial latent to the final image is shown in Figure 10. We further illustrate the specific embedding and extraction procedures between the message and the watermarked latent in the following discussion.

Embedding. In Figure 11, the latent vector is partitioned according to the magnitudes of its coordinates. The green and orange regions correspond to the selected coordinates used for watermark embedding, while the blue region contains the remaining coordinates that are not modified. The selected coordinates are mapped to the first part of the channel, where their signs are adjusted according to the watermark bits. Afterward, a global permutation is applied to obtain the final watermarked latent.

Extraction. Figure 12 illustrates the reverse process. The latent is first inverse-permuted to restore the canonical ordering. The watermark bits are then extracted from the signs of the coordinates in the first part of the channel.

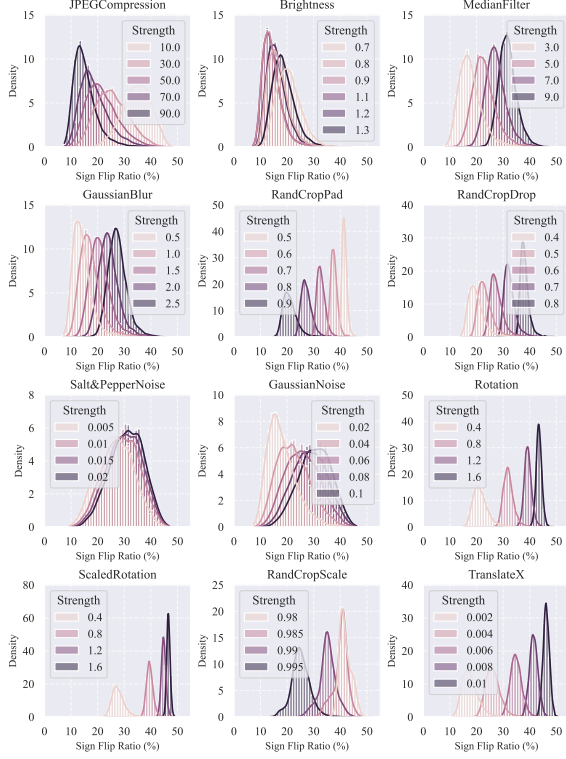


Figure 9: Ratio of latent sign flips induced by different image perturbations.

J Example Images of Different Schemes

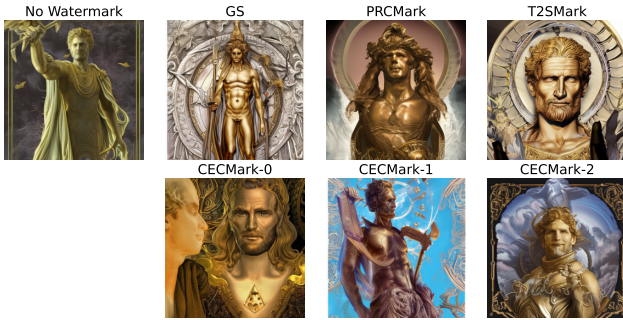


Figure 13: Images generated with prompt “*ultra realistic illustration, a golden statue of a herculean glenn howerton as the god apollo, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by artgerm and greg rutkowski and alphonse mucha*” under different schemes.

K Details of Empirical Distinguishability

Experiment Settings. For each of the 1,000 prompts, we generate 9 watermarked and 9 unwatermarked images for training, and 1 watermarked and 1 unwatermarked image for

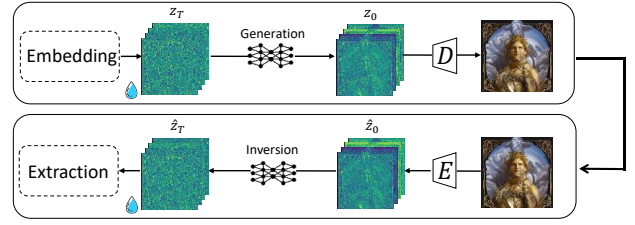


Figure 10: Overview of the generation and inversion processes between the initial latent and the image. The latent consists of four channels. Generation and inversion are implemented using U-Net, while D-Net ($\mathcal{D}$) and E-Net ($\mathcal{E}$) denote the decoder and encoder networks, respectively.

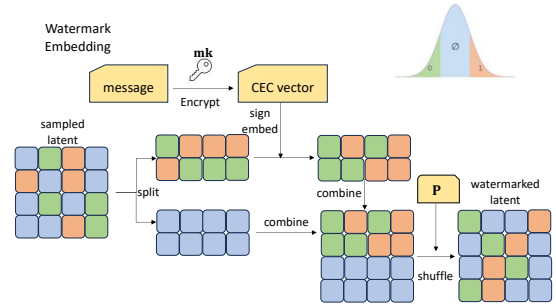


Figure 11: Watermark embedding of CECMark. The green, blue, and orange areas represent positions with large negative, near-zero, and large positive values, respectively. The watermark is only embedded into regions with high magnitude, i.e., the green and orange areas.

testing, all using the same key. This results in 18,000 training samples and 2,000 testing samples. Each model is trained from the pre-trained ResNet-18 model for 10 epochs with a batch size of 64, using the Adam optimizer at a learning rate of 10^{-5} .

Experiment Result. The loss and accuracy on the training and testing datasets for each scheme are shown in Figure 14. GS watermarked images can be distinguished with near-perfect accuracy, and T2SMark also reaches over 80% distinguishability, indicating these schemes have limited confidentiality. In contrast, CECMark-0 achieves the lowest distinction accuracy, which remains below 60%. CECMark-1 is slightly more distinguishable than PRCMark but still remains around 60%. However, CECMark-2 exhibits only moderate confidentiality, with distinguishability rising to about 70%. This increase in distinguishability among the CECMark variants may be attributed to the mixture of the Conc-CEC code and true randomness, which introduces a larger distributional shift and makes the watermark more distinguishable.

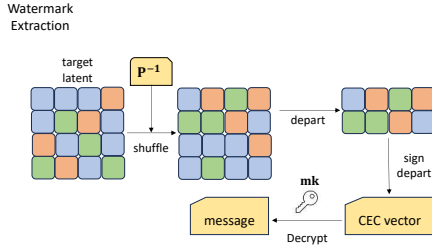


Figure 12: Watermark extraction of CECMark. After inverting the shuffle, the watermark bits are directly extracted as the signs of the first n positions.

L Instantiability of PRC

In this part, we present a comprehensive analysis of the limitations of all currently known PRC instances. We divide the current PRCs into two categories: secret-key PRCs and public-key PRCs.

L.1 Secret-key PRC

The first category is the secret-key version, including PRF-based PRCs [16, 31] and spaced-bounded PRCs in [16].

- **Local PRFs-Based PRC [31].** The core idea of this instance is to embed a pattern that links the input and output of a PRF, specifically $(\mathbf{x} \| F(\mathbf{x}) + \mathbf{e})$, where $F: \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}$ is a local PRF depending only on $\Omega(\log(n))$ bits of the input, and $\mathbf{e}$ represents a Bernoulli error. However, this locality introduces a significant vulnerability to statistical analysis. An adversary against undetectability can collect $O(n^2)$ samples $\{\mathbf{y}^{(j)}\}$ and examine the joint distribution of coordinate pairs $(\mathbf{y}_\alpha^{(j)}, \mathbf{y}_\beta^{(j)})$ for all indices α, β . While these pairs are statistically independent in a truly uniform distribution, the watermarked distribution exhibits a detectable bias. Specifically, if $\mathbf{y}_\alpha$ corresponds to an input bit of F and $\mathbf{y}_\beta$ to its output, the conditional probability satisfies:

$$\Pr[\mathbf{y}_\beta^{(j)} = \mathbf{y}_\beta^{(k)} \mid \mathbf{y}_\alpha^{(j)} = \mathbf{y}_\alpha^{(k)}] \geq \frac{1}{2} + \frac{1}{2\Omega(\log n)} = \frac{1}{2} + \frac{1}{\Omega(n)}$$

where the bias $\frac{1}{\Omega(n)}$ arises whenever the $\Omega(\log n)$ bits influencing F happen to coincide. This indicates that while local PRFs offer efficiency, they fail to provide robust undetectability against polynomial-time distinguishers.

- **PRF-Based PRC [16].** A different approach is taken in [16], which proposes a zero-bit construction utilizing a PRF with a larger input domain of $O(n)$ instead of $\Omega(\log n)$ to mitigate such local correlations. Although

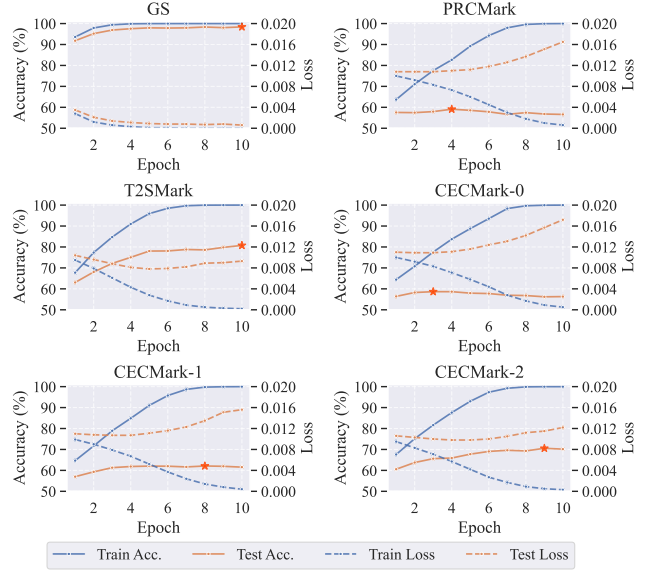


Figure 14: The loss and accuracy curves for ResNet-18 classifiers trained to distinguish watermarked from unwatermarked images.

this design achieves stronger undetectability, the scheme can only tolerate *sub-constant error rates*, rendering it impractical for noisy real-world channels.

- **Space-bounded PRC [16].** In Section 4 of [16], the authors propose a space-bounded PRC of the form $(\mathbf{a} \| \mathbf{a} \cdot \mathbf{s}_1 + \mathbf{e}_1 \| \dots \| \mathbf{a} \cdot \mathbf{s}_k + \mathbf{e}_k)$, where $\{\mathbf{s}_i\}$ are sparse vectors and $\{\mathbf{e}_i\}$ are Bernoulli errors. Due to the sparsity of $\{\mathbf{s}_i\}$, this construction remains vulnerable to statistical analysis. Accordingly, the authors only claim undetectability against adversaries with $O(n^{1.5})$ space complexity, which limits its practical security guarantees.

L.2 Public-key PRC

The second category comprises public-key constructions that typically rely on hardness assumptions [8, 9, 16, 19].

- **Planted XOR-Based PRC [9, 16, 19].** Schemes in [9, 19] and Section 5 of [16] are based on the planted XOR problem. These constructions are known to be vulnerable to quasi-polynomial-time attacks.
- **Planted Hyper-loop-Based PRC [16].** The construction in Section 4 of [16] relies on the planted hyper-loop problem, which is also susceptible to quasi-polynomial-time attacks.
- **Reed-Solomon-Based PRC [8].** Recent attempts in [8] employ Reed-Solomon codes to construct PRCs, but exhibit significant theoretical and practical shortcomings. Specifically, the construction in Theorem 5.6 is limited

to deletion channels, while the construction in Theorem 5.7 requires a decoder with exponential-time complexity, making it computationally infeasible. As a result, these constructions do not fully meet the PRC requirements in their current form.

Crucially, most of these works lack concrete parameter instantiations, remaining purely theoretical. Even for the scheme in [19], which does provide parameters, recent cryptanalysis in [39] demonstrates that its undetectability can be compromised by a polynomial-time adversary. Consequently, existing PRCs remain far from practical deployment.

M Attacks against Confidentiality of PRC-Mark

We further adopt Attack-I in [39] to break the confidentiality of PRCMark. Specifically, we first recover l rows of the secret key $\mathbf{P}$ using Attack-I, and construct a partial secret key $\mathbf{P}' \in \mathbb{F}_2^{l \times n}$. Then, we apply the Decode() algorithm exactly as in [19] to decode the message.

To verify the effectiveness of our attack, we apply PRCMark with $t = 3$ on stable-diffusion-2-1-base [33]. We randomly generate 100 pairs of $\mathbf{pk} = (\mathbf{G}, \mathbf{z})$. For each $\mathbf{pk}$, we generate 100 images. Given each public key $\mathbf{pk}$, we recover a partial secret key $\mathbf{P}'$ by setting $l = 10,000$. The results show that we successfully recover 97.99% of the messages, indicating that the confidentiality of PRCMark is not satisfied.