Advancing Pseudorandom Codes: Beyond Parity Checks and Standard-Model CCA1 Security

Yu Chen[1], Xinyu Mao[2], and Hongxu Yi[1]

[1] School of Cyber Science and Technology, Shandong University, Qingdao 266237, China
yuchen@sdu.edu.cn, tcs.hongxu.yi@mail.sdu.edu.cn
[2] Thomas Lord Department of Computer Science, University of Southern California
xinyumao@usc.edu

Abstract. Pseudorandom codes (PRCs) are error-correcting codes whose codewords are computationally indistinguishable from uniform random strings, a primitive motivated by the need to robustly watermark generative AI models. While recent breakthroughs have established the feasibility of PRCs, critical challenges remain regarding the diversity of their underlying cryptographic assumptions and their security against active adversaries. This work advances the study of PRCs on two complementary fronts: structural diversity and advanced security.

On the structural side, we propose a novel PRC template that departs from the prior one based on sparse parity-check trapdoors. We introduce a new LPN-type assumption, formalized as Dense-Planted LPN, which postulates $(\mathbf{MT},\ \mathbf{MTs}+\mathbf{e}) \approx_c (\mathbf{MT},\ \mathbf{u})$, where $\mathbf{T}$ is a random dense matrix and $\mathbf{M}$ is sampled from a distribution containing a planted structure. This hidden structure enables a completely new decoding mechanism based on a local-window search rather than global parity checks. Notably, this template provides a viable path toward constructing PRCs from assumptions beyond code-based ones, including Learning with Errors (LWE) assumptions.

On the security side, we construct the first public-key PRC secure against pre-challenge chosen-codeword attacks (CCA1) in the standard model. In realistic watermarking deployments where detectors are exposed as public services, CCA security is essential. However, prior CCA-secure PRCs were only achievable in the random oracle model. By formally introducing and instantiating a robust tag-based equivocal bit commitment scheme combined with robust hinting PRGs, we demonstrate that CCA1 security can be achieved in the standard model without sacrificing decoding robustness.

1 Introduction

For codes, error correction and pseudorandomness are inherently in delicate tension. Error-correction relies on structure, allowing a decoder to recognize and recover codewords even after they are corrupted by noise. Pseudorandomness, conversely, demands the complete absence of detectable structure: to an efficient

adversary without the secret key, codewords should be indistinguishable from a uniformly random string. The notion of a ***pseudorandom code*** (PRC), introduced by Christ and Gunn [CG24], elegantly reconciles this paradox. A PRC is an error-correcting code whose encodings are computationally indistinguishable from random, yet a legitimate decoder possessing the secret key can still recover or detect the encoded message even under adversarial corruptions.

The study of PRCs has accelerated rapidly, driven in large part by its compelling application to the watermarking of generative AI models. For instance, recent work by Gunn, Zhao, and Song [GZS25] demonstrated the practical deployment of PRCs to watermark generative image models. In these applications, pseudorandomness ensures the watermark remains undetectable and thus preserves generation quality, while error correction provides robustness against adversaries attempting to remove the watermark. Consequently, a flurry of recent works has constructed PRCs from LPN-type assumptions, planted-combinatorial assumptions, and weak PRFs, while exploring stronger robustness properties [CG24, GG24, MG24, AAC+25, CGG+25].

Despite this rapid progress, the theoretical foundation of PRCs remains constrained by two critical challenges:

The Structural Frontier: Moving Beyond Parity Checks. Currently, the design of public-key PRCs is tightly coupled to code-based noisy-linear assumptions. Crucially, the decoding mechanisms in existing constructions rely almost exclusively on sparse parity-check trapdoors. This theoretical bottleneck raises an intriguing question:

Can we build PRCs from entirely different trapdoor mechanisms?

Diversifying the structural approach is vital, as it represents the most promising path toward constructing PRCs from assumptions beyond traditional code-based cryptography, such as Learning with Errors (LWE).

The Security Frontier: CCA Security in the Standard Model. In the basic PRC pseudorandomness experiment, the adversary can only launch chosen-plaintext attacks it sees encodings but lacks access to a decoder. Nevertheless, in real-world watermarking and detection settings, AI watermark detectors are routinely exposed as public APIs, granting adversaries pre-challenge access to a detection oracle. Chosen-codeword attacks (CCA) for PRCs were recently formalized by Alrabiah et al. [AAC+25], but their constructions rely heavily on the random oracle model (ROM). The following problem remains open:

How to attain CCA security for PRCs in the standard model?

This work makes decisive progress on both fronts. We introduce a novel LPN-type assumption—Dense-Planted LPN to build a PRC whose decoder relies on a hidden local window rather than a global parity check. Furthermore, we construct the first standard-model PRC secure against pre-challenge chosen-codeword attacks (CCA1) from standard assumptions.

1.1 Our Contributions

We detail our two main contributions below.

A New Paradigm for PRC: Dense-Planted Assumption and Local Decoding. We propose a fundamentally new approach to PRC design, departing from sparse parity-check trapdoors. To do so, we introduce the Dense-Planted LPN assumption. This assumption is inspired by the Dense-Sparse LPN assumption [DJ24] and the unbalanced expansion assumption [ABW10]. Let $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$ be the distribution over sparse matrices $\mathbf{M} \in \mathbb{F}_2^{m \times N}$ obtained as follows. First, sample each row as a uniformly random vector of Hamming weight d. Then embed a planted structure: choose hidden sets $I \subseteq [m]$ and $J \subseteq [N]$, with $|I| = Q$ and $|J| = t = \lfloor Q/3 \rfloor$, and resample every row indexed by I so that its support lies entirely inside J. The planted sets (I, J) act as a hidden trapdoor and are kept secret.

The Dense-Planted LPN assumption states that

$$\big(\mathbf{MT},\ \mathbf{MTs} + \mathbf{e}\big) \approx_{\mathsf{c}} \big(\mathbf{MT},\ \mathbf{u}\big),$$

where $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$, $\mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}$ is a dense random matrix, and $\mathbf{s} \leftarrow \mathbb{F}_2^r$, $\mathbf{e} \leftarrow \mathrm{Ber}(m, \eta)$, $\mathbf{u} \leftarrow \mathbb{F}_2^m$. From this assumption, we construct a zero-bit public-key PRC.

Theorem 1 (Informal). *Let $\delta \in (0, 1/10)$ be any constant. Under the Dense-Planted LPN assumption, there exists a zero-bit public-key PRC that is robust to every δ-bounded channel.*

The local decoding template. In our construction, key generation samples $\mathbf{M}$ together with the hidden window (I, J), samples $\mathbf{T}$, sets $\mathbf{A} = \mathbf{MT}$, and samples a public shift $\mathbf{z}$. The public key is $(\mathbf{A}, \mathbf{z})$, while the secret key contains $(I, J, \mathbf{M}_{I,J}, \mathbf{z})$. An encoding is

$$\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{z}, \tag{1}$$

where $\mathbf{s} \leftarrow \mathbb{F}_2^r$ and $\mathbf{e} \leftarrow \mathrm{Ber}(m, \eta)$. To decode, rather than computing a global parity check, the decoder restricts the received word to the hidden row set I. For honest codewords, this local view has the form

$$(\mathbf{c} - \mathbf{z})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J + \mathbf{e}_I.$$

Because the unknown vector has only t coordinates (where $t = O(logN)$ in the canonical regime), the decoder can efficiently brute-force all $N = 2^t$ candidates and accept if one is close to the local word.

This establishes a new design template for PRC. The public key is a dense, noisy linear object, while the secret key stores a tiny planted block that enables brute-force decoding because its dimension is logarithmic.

A conditional Dense-Planted LWE template. To demonstrate the versatility of our local-window template, we also formulate the natural LWE analogue over $\mathbb{Z}_q$. Here, the public matrix remains $\mathbf{A} = \mathbf{MT}$, and codewords take the form

$$\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{z} \pmod{q},$$

where $\mathbf{e}$ is an LWE error vector. The same hidden window gives the decoder a local instance of noisy linear algebra over $\mathbb{Z}_q$.

Transitioning from LPN to LWE introduces a significant new hurdle: decoding under bit-level corruptions. A binary corruption of the bit-decomposition of a q-ary word does not simply add another short error term. To successfully decode, one must solve the Mixed Error Decoding (MED) problem—namely, deciding whether a vector $\mathbf{y}$ can be written as $\mathbf{Cv} + \mathbf{e}_0 + \mathbf{Ge}_1 \pmod{q}$, where $\mathbf{e}_0$ is small and $\mathbf{e}_1$ is sparse.

Theorem 2 (Informal). *Let $\delta \in (0, 1/10)$ be any constant. Assume that MED instances of dimension Q can be solved in $2^{O(Q)} \cdot \mathtt{poly}(q) = \mathtt{poly}(N)$ time. Then, under the Dense-Planted LWE assumption, there exists a zero-bit public-key PRC that is robust to every δ-bounded channel.*

This result is meant as a template rather than a complete LWE-based construction. It successfully separates the cryptographic assumption from the local algorithmic problem required for decoding. We also connect the MED problem to the recently proposed Learning-with-Two-Errors (LW2E) assumption [GJL+25], framing a concrete algorithmic challenge for future research.

The First Standard-Model CCA1-Secure PRC. Our second major contribution resolves the open problem of achieving chosen-codeword security for PRCs without relying on random oracles. We formally revisit the definition of CCA security for PRCs and provide a standard-model construction secure under a pre-challenge chosen-codeword attack (CCA1), where the adversary is granted decoder access prior to receiving the challenge codeword.

Theorem 3. *Assume (1) the subexponential LPN assumption and (2) the LWE assumption (or the computational Diffie-Hellman assumption) hold. Then there exists a standard model PRC that encodes λ-bit messages, is CCA1-pseudorandom, and satisfies adaptive* 0.0005*-robustness with decoder access.*

To achieve this, we avoid generic NIZK-based paradigms (like Naor-Yung), which are difficult to make both robust and pseudorandom. Instead, we adapt the "signaling technique" and hinting PRGs from the public-key encryption literature [KW19, GHMO21]. The technical core of our result is the conceptualization and construction of a Robust Tag-Based Equivocal Bit Commitment (Robust TBEQC) scheme. This primitive is carefully designed so that the verification process can tolerate noise on both the tag and the commitment string, allowing us to bootstrap CPA-secure PRCs into CCA1-secure PRCs while fully preserving error-correction capabilities.

1.2 Technique Overview

We next give the main ideas behind the constructions and proofs.

PRC via Local Decoding The construction samples $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$ and $\mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}$ as in the Dense-Planted LPN assumption, and publishes $\mathbf{A} = \mathbf{MT}$ as public key. The secret key is the hidden sets $I \subseteq [m]$ and $J \subseteq [N]$, with $|I| = Q$ and $|J| = t = Q/3$, such that every row of $\mathbf{M}$ indexed by I is supported inside J.

A codeword is of the form $\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{z}$ as in Equation (1). Given a corrupted word $\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{f}$, where $|\mathbf{f}| < \delta m$ and δ is our targeted PRC error rate. With secret key (I, J), one can compute $\mathbf{z}$. The local word is

$$(\widetilde{\mathbf{c}} - \mathbf{z})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J + \mathbf{e}_I + \mathbf{f}_I.$$

Since $(\mathbf{Ts})_J \in \mathbb{F}_2^t$ and $t = O(\log N)$, the decoder can efficiently enumerate every candidate $\mathbf{v} \in \mathbb{F}_2^t$ and check whether $\mathbf{M}_{I,J}\mathbf{v}$ is close to the local word, say, within Hamming distance $0.15Q$. Note that decoding uses a brute-force search over a local block rather than a global parity check.

Robustness, soundness, and pseudorandomness of the local test. Robustness follows from a concentration bound. If the channel introduces a global error vector of relative weight δ, and if the error locations are independent of the hidden set I, then the number of external errors landing in I is hypergeometric with mean pQ. The internal LPN noise contributes another ηQ errors in expectation. When $\delta + \eta < 1/8$, $|\mathbf{e}| + |\mathbf{f}| < 0.15Q$ except with inverse-polynomial probability. This weak guarantee can be amplified to get standard robustness.

Soundness relies on the public shift $\mathbf{z}$. For any fixed word $\mathbf{u}$, the local vector $(\mathbf{u}-\mathbf{z})_I$ is uniform over $\mathbb{F}_2^Q$. The decoder accepts only if this uniform vector lands in one of 2^t Hamming balls of radius $0.15Q$. Since $t = Q/3$ and $1-\mathbf{H}(0.15) > 1/3$, a union bound gives inverse-polynomial false-accept probability.

Pseudorandomness readily follows from the Dense-Planted LPN assumption: for polynomially many encoding queries, the response matrix has the form

$$\mathbf{AS} + \mathbf{E} + \mathbf{z1}^T = \mathbf{MTS} + \mathbf{E} + \mathbf{z1}^T,$$

and the matrix form of Dense-Planted LPN replaces $\mathbf{MTS} + \mathbf{E}$ by a uniform matrix.

CCA1-Secure PRC Construction We begin with the KW construction for obtaining CCA [3] security from chosen plaintext security using hinting PRG [KW19], specialized to the form that underlies our construction.

Let $\Pi = (\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ be a PRC that encodes multi-bit messages. Two core ingredients are a *tag-based bit commitment scheme* $\mathcal{C}$ and a

[3] For encryption scheme CCA stands for chosen ciphertext attack. We still use the abbreviation CCA when the context is clear.

pseudorandom generator (PRG) $\mathsf{PRG} : \{0,1\}^N \to (\{0,1\}^\ell)^N$. The ciphertext is organized into a core component and N blocks. Our CCA1-secure PRC uses $2N$ key pairs of Π: $\{(pk_{b,i}, sk_{b,i})\}_{b\in\{0,1\},i\in[N]}$. A message m is encoded as follows.

1. Sample a seed $s \leftarrow \{0,1\}^N$ and compute $\mathsf{PRG}(s) = (y_0, \ldots, y_N)$.
2. (Core component) $\mathsf{ct}_0 := y_0 \oplus m$.
3. For each coordinate $i \in [N]$, one samples a tag t_i, use $\mathcal{C}$ to commit to the bit s_i with tag t_i, and creates two Π-codeword components. The active branch $b = s_i$ encodes a valid opening ρ_i to the commitment com_i with randomness y_i:
$$c_{s_i,i} = \mathsf{Encode}(pk_{s_i,i}, 1\|\rho_i; y_i),$$
whereas the inactive branch is generated using fresh randomness,
$$c_{1-s_i,i} = \mathsf{Encode}(pk_{1-s_i,i}, 0^{\ell_{msg}}).$$
Thus, each block has the form $\mathsf{ct}_i := t_i\|\mathsf{com}_i\|c_{0,i}\|c_{1,i}$,
4. Output the concatenated codeword $\mathsf{ct} = \mathsf{ct}_0\|\mathsf{ct}_1\|\cdots\|\mathsf{ct}_N$.

The CCA security of the KW construction relies on additional properties of $\mathcal{C}$ and PRG:

- The commitment scheme has an *equivocal mode*, where for some challenge tag t^*, one can produce a commitment com with two valid openings: ρ_0 for the bit 0 and ρ_1 for the bit 1. This is formalized as *tag-based equivocal commitment (TBEQC)* in [Mat25].
- The PRG is a *hinting PRG*, namely, it satisfies the following hinting property: for a uniform random seed s and $(y_0, \ldots, y_N) := \mathsf{PRG}(s)$,
$$y_0, (z_{b,i})_{b\in\{0,1\},i\in[N]} \text{ is pseudorandom, where } z_{s_i,i} := y_i,\ z_{1-s_i,i} \leftarrow \{0,1\}^\ell.$$

Make the scheme robust. Decoding is easy if there is no noise: one first recovers the seed s by decoding $(c_{0,i}, c_{1,i})$ for each i to see which branch encodes a valid opening for com_i with tag t_i. Then it computes $(y_0, \ldots, y_N) = \mathsf{PRG}(s)$, recovers $m = \mathsf{ct}_0 \oplus y_0$, and performs re-encryption checks for $(c_{s_i,i})_{i\in[N]}$. In the following, we outline our adaptations to make the scheme robust against corruption.

- With suitable parameter settings, one can assume only αN blocks are corrupted beyond the decoding capacity of Π for some constant α. The hope is that we can still recover $(1-\alpha)N$ bits of s. The difficulty is that we also need to check whether the underlying message contains a valid opening for com_i with tag t_i, but now both the tag and the commitment could be corrupted. To address this issue, we formalize *robust TBEQC* to replace TBEQC and construct it from one-way functions.
- Now suppose that we can recover $(1-\alpha)N$ bits of s_i. Our second adaptation is that, instead of sampling $s \leftarrow \{0,1\}^N$, we choose $\sigma \leftarrow \{0,1\}^\kappa$ and set the PRG seed to be $s := \mathsf{ECC}(\sigma) \in \{0,1\}^N$ for some error-correction code ECC that corrects up to an α fraction of errors. We need the hinting

property to hold even if the seed s is not uniform but has sufficient entropy. This is exactly the notion of *robust hinting PRG* considered by Garg, Hajiabadi, Malavolta, and Ostrovsky [GHMO21], and they observed that several constructions of hinting PRGs can be adapted to obtain robust ones.
– Finally, to make ct_0 robust, we set $\mathsf{ct}_0 := y_0 \oplus \mathsf{ECC}'(m)$ for some properly chosen error correction code ECC'. Once we recover σ and recompute $(y_0, \ldots, y_N) := \mathsf{PRG}(\mathsf{ECC}(\sigma))$, we can rely on ECC' to decode m from $\widehat{\mathsf{ct}}_0 \oplus y_0$ and conduct re-encryption checks as in the KW construction, where $\widehat{\mathsf{ct}}_0$ is a corrupted version of ct_0.

Why not CCA2 security? The major obstacle is robust TBEQC. In fact, we can get CCA2 security in the following restricted sense: Let us use the same tag t for all blocks and suppose that (1) no corruptions are allowed on the tag t, and that (2) the decoding oracle rejects queries with tag equal to the challenge tag t^*. In the context of public-key encryption, this is known as *tag-based encryption with weak CCA security*, and it can be converted to a CCA2-secure PKE generically [Kil06,MH15]. However, the conversions either introduce additional components that are not pseudorandom or seem difficult to adapt to tolerate corruptions on the tag.

We achieve CCA1 security by constructing a robust TBEQC such that the verification can tolerate corruptions on both the tag and the commitment, with the following constraint: the tag t_i must be reasonably far from the challenge tag t_i^* in the sense that $t_i \oplus t_i^*$ is balanced: $\gamma n_{tag} \leq |t_i \oplus t_i^*| \leq (1-\gamma) n_{tag}$ where n_{tag} is the tag length and $\gamma \in (0, 1/2)$ is a constant. In the CCA1 model, t_i^* and t_i are independent, and hence this holds with overwhelming probability.

1.3 Related Work and Discussion

Pseudorandom codes and watermarking. Early watermarking schemes for generative AI, most notably the text watermarking framework by Christ, Gunn, and Zamir [CGZ24], succeeded in ensuring undetectability but lacked robustness against adversarial corruptions. To resolve this fundamental tension between quality preservation and robustness, Christ and Gunn [CG24] formally introduced the primitive of pseudorandom codes. They provided two initial public-key PRC constructions based on either the subexponential LPN assumption or a combination of LPN with a planted XOR assumption, thereby establishing the first framework for quality-preserving and robust watermarks. Following this breakthrough, Gunn, Zhao, and Song [GZS25] demonstrated the practical deployment of PRCs in watermarking generative image models, while a sequence of subsequent works expanded the frontier toward stronger robustness guarantees [AAC+25,CGG+25] and different noisy channels [GM24,MG24]. Within this rapidly growing literature, Ghentiyala and Guruswami [GG24] established robustness amplification frameworks for PRCs, while noting that defending against an $o(1)$ corruption rate can be trivially achieved. This motivates the subsequent focus on the constant-error regime. The question of whether pseudorandomness can be amplified remains open.

Subexponential security. It is an intriguing open question to construct a public-key PRC with subexponential security. All existing public-key PRC constructions, including ours, are vulnerable to quasipolynomial-time attacks, while a recent secret-key PRC construction achieves subexponential security under non-standard yet plausible assumptions.

PRC beyond code-based assumption. Constructing PRCs beyond code-based assumptions faces theoretical barriers. Recent negative results [GGW25, DMR25] demonstrate that even with black-box access to powerful primitives like indistinguishability obfuscation (iO), there are no black-box constructions of PRCs. Our work makes the first attempt to explore LWE-type directions for PRC design. While our Dense-Planted LWE template does not yet yield a complete PRC due to the requirement of an efficient solver of the Mixed Error Decoding problem, which is closely related to the recent LW2E assumption [GJL+25]. This leaves an explicit algorithmic problem for future research.

Chosen codeword security. Alrabiah et al. initiated the study of chosen-codeword security (CCA) for PRCs, and gave generic constructions in the random oracle model inspired by the Fujisaki-Okamoto transformation [AAC+25, FO99]. For public-key encryption, achieving chosen-ciphertext security in the standard model is well established: For example, Naor-Yung paradigm [NY90,Sah99] gives a generic CPA-to-CCA transformation that uses NIZK proof to enforce ciphertext well-formedness, and CCA security can also be obtained from lossy cryptography [PW08]. Our approach follows the line of research that achieves CCA security via circular security and the so-called 'signaling technique ' [HKW20,KW19, MH14, GGH19, GHMO21]. We choose this approach because its ciphertext components appear amenable to being made both robust and pseudorandom. For example, it seems difficult to obtain a robust, pseudorandom NIZK proof if one starts with the Naor-Yung paradigm. Moreover, many transformations follow the paradigm of obtaining CCA2 security from a tag-based encryption with weak CCA security. However, the simple transformation using one-time digital signature [Kil06] does not retain pseudorandomness.

2 Preliminary

For $n \in \mathbb{N}$, we write $[n] = \{1, \ldots, n\}$. We use $\leftarrow$ to denote sampling from a distribution, choosing an element from a set uniformly at random, or collecting the output of a randomized algorithm.

For a vector $\mathbf{x}$, let $|x| \stackrel{\mathsf{def}}{=} |\{i : x_i \neq 0\}|$ denote its Hamming weight, and write $\Delta(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$. For $0 \leq t \leq n$, define the Hamming sphere $\mathcal{S}_{t,n} = \{\mathbf{x} \in \mathbb{F}_2^n : |\mathbf{x}| = t\}$.

For a matrix $\mathbf{M}$, we write $\mathbf{M}_{I,J}$ for the submatrix indexed by rows I and columns J. We use $\mathrm{Ber}(n, p)$ for the length-n product Bernoulli distribution with marginal error rate p, and $\mathrm{Ber}(n \times m, p)$ for matrix distribution over $\mathbb{F}^{n \times m}$ where each entry is sampled from $\mathrm{Ber}(n, p)$ independently.

A function $\mu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if $\mu(\lambda) = \lambda^{-\omega(1)}$, denoted $\mu(\lambda) = \mathtt{negl}(\lambda)$.

2.1 Basic Concentration Bounds

Lemma 1 (Hamming ball bound). *Let $V_2(n,r) = \sum_{i=0}^{r} \binom{n}{i}$. If $0 < r \leq n/2$ and $p = r/n$, then $V_2(n,r) \leq 2^{nH(p)}$, where $\mathbf{H}(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function.*

Lemma 2 (Hoeffding bound). *Let $X_1, \ldots, X_n$ be independent random variables in $[0,1]$, and let $S = \sum_i X_i$. If $\mu = \mathbf{E}[S]/n$, then for every $\epsilon > 0$, $\mathbf{Pr}[S \geq (\mu+\epsilon)n] \leq \exp(-2\epsilon^2 n)$.*

Lemma 3 (Union bound). *For events $A_1, \ldots, A_n$, one has $\mathbf{Pr}[\bigcup_i A_i] \leq \sum_i \mathbf{Pr}[A_i]$.*

Lemma 4 (Hypergeometric tail). *Let $X \sim \mathrm{Hyp}(N,K,n)$, and put $p = K/N$. For every $\epsilon > 0$, $\mathbf{Pr}[X \geq (p+\epsilon)n] \leq \exp(-2\epsilon^2 n)$ and $\mathbf{Pr}[X \leq (p-\epsilon)n] \leq \exp(-2\epsilon^2 n)$.*

Lemma 5 (Adding Bernoulli noise). *Let $\mathbf{w} \in \mathbb{F}_2^n$ be fixed with relative weight at most $p_w < 1/2$, and let $\mathbf{x} \leftarrow \mathrm{Ber}(n,\eta)$ for $\eta < 1/2$. Then $\mathbf{E}[|\mathbf{w}+\mathbf{x}|]/n = p_w + \eta - 2p_w\eta \leq p_w + \eta$. Moreover, for every constant $\epsilon > 0$, the probability that $|\mathbf{w}+\mathbf{x}|$ exceeds its expectation by ϵn is at most $\exp(-2\epsilon^2 n)$.*

Proof. Let $Y_i = w_i \oplus x_i$ for $i \in [n]$ be independent indicator variables. Since $x_i \leftarrow \mathrm{Ber}(\eta)$, their expected value is $\mathbf{E}[Y_i] = w_i + \eta - 2w_i\eta$. By the linearity of expectation and the fact that $|\mathbf{w}| \leq p_w n$, we have $\mathbf{E}[|\mathbf{w}+\mathbf{x}|]/n = p_w + \eta - 2p_w\eta \leq p_w + \eta$. Since $Y_1, \ldots, Y_n$ are independent variables in $[0,1]$, directly applying the Hoeffding bound yields: $\mathbf{Pr}\left[\sum_{i=1}^{n} Y_i \geq (\mu+\epsilon)n\right] \leq \exp(-2\epsilon^2 n)$.

2.2 Pseudorandom Codes

Communication channel. Given alphabet Σ, a ***channel*** is a map $\mathcal{E}$: $\Sigma^* \to \Sigma^*$ that maps a string Σ^* to a distribution over Σ^*. We say that a length-preserving channel $\mathcal{E}$ is δ***-bounded*** if for all $x \in \Sigma^n$, the output distribution stays within a Hamming distance of δn from the input with overwhelming probability. Formally, it holds that $\mathbf{Pr}_{y \leftarrow \mathcal{E}(x)}\left[\Delta(x,y) > \delta n\right] \leq \mathtt{negl}(\lambda)$.

Definition 1 (PRC [CG24]). *Let $\mathfrak{E} = (\mathfrak{E}_\lambda)_{\lambda \in \mathbb{N}}$ be a class of channels over Σ. A* ***secret-key PRC over alphabet Σ with $\mathfrak{E}$-robustness*** *is a tuple of PPT algorithms $(\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ with the following syntax and properties.*

- *Syntax. Let $\ell_p k, \ell_s k, n, k$ be length functions.*
  - $\mathsf{KeyGen}(1^\lambda) \mapsto (pk, sk)$*: On input security parameter, it outputs a public key $pk \in \{0,1\}^{\ell_{pk}}$ and a secret key $sk \in \{0,1\}^{\ell_{sk}}$.*
  - $\mathsf{Encode}(pk, m \in \Sigma^k) \mapsto c \in \Sigma^n$*: It takes as inputs a public key pk and message m, and outputs a codeword c.*

- $\mathsf{Decode}(sk, c \in \Sigma^n) \mapsto m \in \Sigma^k \cup \{\perp\}$: *It takes as inputs a secret key* sk *and a word* c*, and outputs* $m \in \Sigma^k$ *if the decoding is successful; otherwise, it outputs* $\perp$ *indicating failure.*

– *Robustness.* *For every message* m *and channel* $\mathcal{E} \in \mathfrak{E}_\lambda$*, it holds that*

$$\mathop{\mathbf{Pr}}_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda),c\leftarrow\mathsf{Encode}(pk,m)}[\mathsf{Decode}(sk,\mathcal{E}(c)) = m] \geq 1 - \mathtt{negl}(\lambda).$$

– *Soundness.* *For any fixed word* $c \in \Sigma^*$*,*

$$\mathop{\mathbf{Pr}}_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Decode}(sk,c) \neq \perp] \leq \mathtt{negl}(\lambda).$$

– *Pseudorandomness.* *For every PPT adversary* $\mathcal{A}$*,*

$$\left|\mathop{\mathbf{Pr}}_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda)}\left[\mathcal{A}^{\mathsf{Encode}(pk,\cdot)}(1^\lambda)\right] - \mathbf{Pr}\left[\mathcal{A}^{\mathcal{U}(\cdot)}(1^\lambda)\right]\right| = \mathtt{negl}(\lambda),$$

where the oracle $\mathcal{U}$ *returns a uniformly random word in* Σ^n *for each query.*

The $\mathfrak{E}$-robustness defined as above is adopted by a range of works on pseudorandom codes (e.g., [CG24, GG24, GM24]). We also adhere to this definition to establish our PRC constructions from new approaches in Section 3 and Section 4.

Alrabiah et al. [AAC+25] consider a stronger robustness notion termed *adaptive robustness*: An adversary sees the public key and wins if it submits a message-randomness pair that witnesses a decoding error. Formally, define the public-key adaptive robustness game $\mathcal{G}^{\mathsf{robust\text{-}pk}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda)$ as follows:

1. The challenger samples $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends pk to the adversary.
2. The adversary sends (m, r, x) to the challenger.
3. The challenger computes $c = \mathsf{Encode}(pk, m; r)$ and $m' = \mathsf{Decode}(sk, x)$. If $\Delta(x, c) \leq \delta n$ and $m' \neq m$, then the adversary **wins**; otherwise the adversary **loses**.

Definition 2 (Adaptive robustness [AAC+25]). *We say that a public-key pseudorandom code* PRC *is adaptively* δ*-robust if, for any efficient adversary* $\mathcal{A}$*,*

$$\mathbf{Pr}\left[\mathcal{A} \text{ wins } \mathcal{G}^{\mathsf{robust\text{-}pk}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda)\right] \leq \mathtt{negl}(\lambda).$$

Consider a setting where a weak PRC scheme satisfies pseudorandomness, but only guarantees a weak variant of robustness and soundness, where valid codewords are correctly decoded with probability $\alpha(\lambda)$, fixed words are falsely accepted with probability $\beta(\lambda)$. While this falls short of a full-fledged PRC, by parallel repetition with a random permutation [GG24], such a weak PRC can be amplified to achieve overwhelming robustness and negligible soundness with mild restrictions.

Lemma 6 (Amplification lemma, adapted from [GG24]). *Assume there exists a weak* PRC *scheme* $(\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ *over alphabet* Σ *that satisfies the following conditions:*

1. ***Weak robustness:*** *For a δ-bounded channel $\mathcal{E}$, and for a $1 - \mathsf{negl}(\lambda)$ fraction of key pairs $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, for every message m, the decoding algorithm correctly decodes the words with probability at least $\alpha(\lambda)$:*

$$\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda), c\leftarrow\mathsf{Encode}(pk,m)} [\mathsf{Decode}(sk, \mathcal{E}(c)) = m] \geq \alpha(\lambda).$$

2. ***Weak soundness:*** *For a $1 - \mathsf{negl}(\lambda)$ fraction of key pairs, for every fixed word c, the decoding algorithm falsely accepts the word with probability at most $\beta(\lambda)$: $\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda)} [\mathsf{Decode}(sk, c) \neq \perp] \leq \beta(\lambda)$.*
3. ***Advantage gap:*** *The probabilities above exhibit a noticeable polynomial gap, i.e., $\alpha(\lambda) - \beta(\lambda) \geq 1/\mathrm{poly}(\lambda)$.*
4. ***Pseudorandomness:*** *The scheme satisfies the pseudorandomness property as defined in definition 1.*

Then, for any constant $\varepsilon > 0$, there exists a corresponding PRC *scheme that achieves overwhelming robustness and negligible soundness against any $(\delta - \varepsilon)$-bounded channel, while preserving its pseudorandomness.*

2.3 Hinting PRG

Definition 3 (Robust hinting PRG). *Let $\kappa, n, \ell_{core}, \ell$ be length functions. A $(\kappa, n, \ell_{core}, \ell)$-**robust hinting PRG** consists of two polynomial-time algorithms $(\mathsf{HPRG.Setup}, \mathsf{HPRG.Eval})$ with the following syntax.*

- $\mathsf{HPRG.Setup}(1^\lambda) \to pp$*: On input the security parameter, it outputs public parameters pp.*
- $\mathsf{HPRG.Eval}(pp, s, i) \to y$*: On input the public parameters pp, an n-bit seed $s \in \{0,1\}^n$, and an index $i \in \{0\} \cup [n]$, it outputs a ℓ_{core}-bit string if $i = 0$, and an ℓ-bit string if $i \in [n]$.*

(κ, n)-Robustness. For any PPT adversary $\mathcal{A}$ and any source distribution $\mathcal{S}$ over $\{0,1\}^n$ with min-entropy at least κ, the adversary's advantage in the following game is negligible:

$$\left| \Pr\left[\mathcal{A}\left(pp, y_0^b, \begin{matrix} y_{1,0}^b, \ldots, y_{n,0}^b \\ y_{1,1}^b, \ldots, y_{n,1}^b \end{matrix} \right) = b \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$$

where the probability is taken over $pp \leftarrow \mathsf{HPRG.Setup}(1^\lambda)$, $s \leftarrow \mathcal{S}$, and $b \leftarrow \{0,1\}$, and the strings are generated as follows:

- *If $b = 0$, $y_0^0 \leftarrow \{0,1\}^{\ell_{core}}$, and for all $i \in [n]$, $x \in \{0,1\}$, $y_{i,x}^0 \leftarrow \{0,1\}^\ell$.*
- *If $b = 1$, $y_0^1 \leftarrow \mathsf{HPRG.Eval}(pp, s, 0)$. For all $i \in [n]$, the active branch is pseudorandom $y_{i,s_i}^1 \leftarrow \mathsf{HPRG.Eval}(pp, s, i)$, while the inactive branch is uniform $y_{i,1-s_i}^1 \leftarrow \{0,1\}^\ell$.*

A robust hinting PRG can be based on the computational Diffie-Hellman (CDH) assumption or the LWE assumption [KW19, GHMO21]:

Theorem 4. *For all $\kappa(\lambda) \geq 3\lambda$ and $n, \ell_{core}, \ell \in \mathtt{poly}(\lambda)$, there exists a $(\kappa, n, \ell_{core}, \ell)$-robust hinting PRG assuming the CDH/LWE assumption.*

3 A New Approach to Constructing PRC

This section presents a novel approach to constructing pseudorandom codes from a new assumption. In Section 3.1, we formalize our new assumption, termed the Dense-Planted LPN assumption, and discuss its plausibility by connecting it to the well-known dense-sparse LPN [DJ24] and decisional unbalanced expansion [ABW10] assumptions. As for cryptanalysis, we discuss several attack strategies against Dense-Planted LPN in Section A.

Section 3.2 presents a (zero-bit) PRC construction based on our new assumption. In Section 3.3, we prove its weak robustness, weak soundness, and pseudorandomness.

3.1 Dense-Planted LPN Assumption

Our assumption is inspired by the dense-sparse LPN assumption proposed by Dao and Jain [DJ24] and the decisional unbalanced expansion assumption introduced by Applebaum, Barak, and Wigderson [ABW10]. Intuitively, it postulates that

$$(\mathbf{MT}, \mathbf{MTs} + \mathbf{e}) \approx_c (\mathbf{MT}, \mathbf{u})$$

for a random (dense) matrix $\mathbf{T}$, and a random matrix $\mathbf{M}$ with planted structure as in the unbalanced expansion assumption.

Definition 4 (Matrix distributions). *Let $m, N, d, Q \in \mathbb{N}$ be parameters such that $d \le t \le N$, where $t := Q/3$. For $d \in \{0, \ldots, N\}$, define a collection of sets $\mathcal{S}_{d,N} := \left\{\mathbf{x} \in \mathbb{F}_2^N : \mathrm{wt}(\mathbf{x}) = d\right\}$. Define two distributions $\mathcal{D}_{\mathsf{rand}}^{m,N,d}$ and $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$ over $\mathbb{F}_2^{m \times N}$ as follows:*

- *The random sparse matrix distribution $\mathcal{D}_{\mathsf{rand}}^{m,N,d}$: Sample $\mathbf{M} \in \mathbb{F}_2^{m \times N}$ by choosing each row $\mathbf{M}_{i,*}$ from $\mathcal{S}_{d,N}$ independently and uniformly.*
- *The planted matrix distribution $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$:*
  1. *Sample $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{rand}}^{m,N,d}$.*
  2. *Sample uniformly random sets $I \subseteq [m], J \subseteq [N]$ such that $|I| = Q, |J| = t = Q/3$.*
  3. *For every $i \in I$, resample the row $\mathbf{M}_{i,*}$ uniformly from*

$$\left\{\mathbf{x} \in \mathbb{F}_2^N : \mathrm{wt}(\mathbf{x}) = d \text{ and } \mathrm{supp}(\mathbf{x}) \subseteq J\right\}$$

  conditioned on no $\sqrt{Q}$ rows being linearly dependent [4].
  4. *Output $\mathbf{M}$.*

  The sets (I, J) are the planted structure and are not included in the public output of the distribution.

[4] Equivalently, we require that $\mathbf{M}_{I,J}$ has dual distance at least $\sqrt{Q}$. This can be done by rejection sampling and is necessary to avoid attacks; see Section A for details.

Assumption 1 (Dense-Planted LPN Assumption) *Let λ be the security parameter. Let $m = m(\lambda), N = N(\lambda), r = r(\lambda), d = d(\lambda), Q = Q(\lambda)$ be polynomially bounded parameters, and let $\eta = \eta(\lambda) \in (0, 1/2)$ be the LPN noise rate. Let $\mathcal{D}_{\mathsf{plant}} := \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$. We say that the (m, N, r, d, Q, η)-Dense-Planted LPN assumption, or $\mathsf{DPLPN}(m, N, r, d, Q, \eta)$ holds if for every PPT adversary $\mathcal{A}$,*

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DPLPN}}(\lambda) \stackrel{\mathsf{def}}{=} |\mathbf{Pr}[\mathcal{A}(\mathbf{MT}, \mathbf{MTs} + \mathbf{e}) = 1] - \mathbf{Pr}[\mathcal{A}(\mathbf{MT}, \mathbf{u}) = 1]| \leq \mathrm{negl}(\lambda),$$

where the probability is taken over

$$\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}, \qquad \mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}, \qquad \mathbf{s} \leftarrow \mathbb{F}_2^r, \quad \mathbf{e} \leftarrow \mathrm{Ber}(m, \eta), \qquad \mathbf{u} \leftarrow \mathbb{F}_2^m.$$

Remark 1 (Typical parameter regime). A typical parameter regime is

$$N = \lambda, \quad m = N^{1.1}, \qquad Q = 3 \log N, \qquad d = \sqrt[4]{\log N},$$

together with $r = \alpha N$ for a constant $\alpha \in (0, 1)$, and a constant noise rate $\eta \in (0, 1/2)$.

Remark 2 (Polynomially many samples). For applications requiring polynomially many pseudorandom codewords, one may use the following matrix-secret form. Let $\ell = \mathrm{poly}(\lambda)$. The corresponding matrix form of Dense-Planted LPN asserts that

$$(\mathbf{MT}, \mathbf{MTS} + \mathbf{E}) \approx_c (\mathbf{MT}, \mathbf{U}),$$

where $\mathbf{S} \leftarrow \mathbb{F}_2^{r \times \ell}, \mathbf{E} \leftarrow \mathrm{Ber}(m \times \ell, \eta), \mathbf{U} \leftarrow \mathbb{F}_2^{m \times \ell}. \mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}, \mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}$. This matrix version follows from the vector version by a standard hybrid argument.

Connection to Dense-Sparse LPN. The Dense-Sparse LPN assumption postulates that for every PPT adversary $\mathcal{A}$,

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{DSLPN}}(\lambda) \stackrel{\mathsf{def}}{=} |\mathbf{Pr}[\mathcal{A}(\mathbf{MT}, \mathbf{MTs} + \mathbf{e}) = 1] - \mathbf{Pr}[\mathcal{A}(\mathbf{MT}, \mathbf{u}) = 1]| \leq \mathrm{negl}(\lambda),$$

where the probability is taken over

$$\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{rand}}^{m,N,d}, \qquad \mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}, \qquad \mathbf{s} \leftarrow \mathbb{F}_2^r, \quad \mathbf{e} \leftarrow \mathrm{Ber}(m, \eta), \qquad \mathbf{u} \leftarrow \mathbb{F}_2^m.$$

The only difference is the distribution of $\mathbf{M}$. Therefore, if we assume

$$\mathbf{MT} \approx_c \mathbf{M'T} \tag{2}$$

where $\mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}, \mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}, \mathbf{M'} \leftarrow \mathcal{D}_{\mathsf{rand}}^{m,N,d}$, then, Dense-Planted LPN follows from Dense-Sparse LPN for $\mathcal{D}_{\mathsf{rand}}^{m,N,d}$. Our noise rate and sparsity parameter d are in a safer regime than that of Dense-Sparse LPN. Dense-Sparse LPN is introduced with the aim of constructing more structured primitives like lossy functions, and hence requires an inverse-polynomial noise rate and a constant

row sparsity [DJ24]. In contrast, our PRC construction operates under a constant noise rate and a super-constant sparsity. Remarkably, this super-constant sparsity ensures that trivial row collisions or low-weight linear dependencies occur with negligible probability. We therefore bypass the need for the specialized, collision-free hypergraph distributions [AK19]. Moreover, constant noise rate makes our assumption resemble the well-established standard LPN assumption [BFKL93], and the super-constant sparsity also supports better hardness of sparse LPN [BBTV25].

The unbalanced expansion assumption states that no PPT adversary can distinguish $\mathcal{D}_{\mathsf{rand}}^{m,N,d}$ from $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$ with $O(1)$ advantage. In our Dense-Planted LPN assumption, the adversary only gets $\mathbf{MT}$, not $\mathbf{M}$. Most attacks on the unbalanced expansion assumption exploit the graph structure of $\mathbf{M}$ by viewing it as a bipartite graph. The intuition of Equation (2) being plausible is that as $\mathbf{T}$ compresses $\mathbf{M}$, it hides the graph structure in $\mathbf{M}$.

3.2 PRC from Dense-Planted LPN

Our local decoding mechanism exploits a dramatic dimension reduction enabled by the planted trapdoor (I, J). By restricting the codeword $\mathbf{c}$ to the secret row indices I, the global equation $\mathbf{c} = \mathbf{MTs} + \mathbf{e}$ reduces to a localized relation $\mathbf{c}_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J + \mathbf{e}_I$. Consequently, the decoder can efficiently perform a brute-force search as the size of I and J is logarithmic.

Construction 1 (Zero-bit PRC from Dense-Planted LPN) *Fix parameters as in Remark 1, and set the local decoding threshold $\tau = 0.15Q$. Define $\Pi_{\mathsf{DP}} = (\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ over alphabet $\mathbb{F}_2$ as follows.*

$\mathsf{KeyGen}(1^\lambda)$**.** *Sample $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$ together with its hidden planted sets (I, J). Sample $\mathbf{T} \leftarrow \mathbb{F}_2^{N\times r}$, set $\mathbf{A} = \mathbf{MT} \in \mathbb{F}_2^{m\times r}$, and sample $\mathbf{z} \leftarrow \mathbb{F}_2^m$. Output $pk = (\mathbf{A}, \mathbf{z})$ and $sk = (I, J, \mathbf{M}_{I,J}, \mathbf{z})$.*

$\mathsf{Encode}(pk)$**.** *Parse $pk = (\mathbf{A}, \mathbf{z})$. Sample $\mathbf{s} \leftarrow \mathbb{F}_2^r$ and $\mathbf{e} \leftarrow \mathrm{Ber}(m, \eta)$. Output the codeword $\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{z} \in \mathbb{F}_2^m$.*

$\mathsf{Decode}(sk, \widetilde{\mathbf{c}})$**.** *Parse $sk = (I, J, \mathbf{C}, \mathbf{z})$, where $\mathbf{C} = \mathbf{M}_{I,J} \in \mathbb{F}_2^{Q\times t}$. Let $\mathbf{y} = (\widetilde{\mathbf{c}} - \mathbf{z})_I$. Enumerate all $\mathbf{v} \in \mathbb{F}_2^t$ and compute $R(\mathbf{v}) = \mathrm{wt}(\mathbf{y} - \mathbf{Cv})$. Output 1 if $\min_{\mathbf{v}} R(\mathbf{v}) \le \tau$, and output $\perp$ otherwise.*

Decoding efficiency. The decoding time is polynomial because $2^t = N$ in the canonical regime. More concretely, the local search costs $O(2^t Qd) = O(N \log^{5/4} N)$ bit operations if the planted block is stored sparsely by tracking the indices of non-zero entries.

3.3 Security Proof

In this subsection, we prove weak robustness against bounded channel noise and weak soundness, both derived from the statistical properties of the localized decoding window. By Lemma 6, these weak guarantees can be subsequently

amplified to yield a full-fledged PRC. Additionally, the pseudorandomness of the construction almost directly follows from the Dense-Planted LPN assumption.

Lemma 7 (Local distribution under channel noise). *Let* $\mathbf{c} \leftarrow \mathsf{Encode}(pk)$*, and let* $\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{f}$*, where* $\mathbf{f}$ *is an error vector of weight* d_{err} *independent of the hidden window* I*. Then the decoder's local view satisfies* $\mathbf{y} = (\widetilde{\mathbf{c}} - \mathbf{z})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J + \mathbf{e}_I + \mathbf{f}_I$*, and* $\mathrm{wt}(\mathbf{f}_I)$ *is distributed as* $\mathrm{Hyp}(m, d_{\mathsf{err}}, Q)$*.*

Proof. Since $\mathbf{c} = \mathbf{MTs} + \mathbf{e} + \mathbf{z}$, subtracting $\mathbf{z}$ and restricting to I gives $(\widetilde{\mathbf{c}} - \mathbf{z})_I = (\mathbf{MTs})_I + \mathbf{e}_I + \mathbf{f}_I$. By the planted support condition, rows in I have support contained in J, hence $(\mathbf{MTs})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J$. Finally, because I is a uniformly random Q-subset independent of $\mathbf{f}$, restricting $\mathbf{f}$ to I is exactly sampling Q locations without replacement from a population with d_{err} errors.

Theorem 5 (Weak robustness). *Let* $\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{f}$*, where* $\mathbf{c} \leftarrow \mathsf{Encode}(pk)$*,* $\mathrm{wt}(\mathbf{f}) = d_{\mathsf{err}} \le \delta m$*, and the error positions are independent of the hidden window* I*. Assume* $\delta + \eta < 1/8$*. For* $\tau = 0.15Q$*, Construction 1 accepts* $\widetilde{\mathbf{c}}$ *with probability* $\alpha(\lambda) \ge 1 - 1/\operatorname{poly}(N)$*. In particular, this gives weak robustness for the* δ*-bounded channel.*

Proof. By Lemma 7, when the decoder enumerates the correct candidate $\mathbf{v}^\star = (\mathbf{Ts})_J$, its residual is $R(\mathbf{v}^\star) = \mathrm{wt}(\mathbf{y} - \mathbf{M}_{I,J}\mathbf{v}^\star) = \mathrm{wt}(\mathbf{e}_I + \mathbf{f}_I)$. The hypergeometric tail bound implies that $\mathrm{wt}(\mathbf{f}_I) \le (\delta + \epsilon_1)Q$ except with probability $N^{-\Omega(1)}$, for every fixed $\epsilon_1 > 0$. Conditioned on this event, fix $\mathbf{f}_I$ and add $\mathbf{e}_I \leftarrow \mathrm{Ber}(Q, \eta)$. By Lemma 5, the expected relative weight of $\mathbf{e}_I + \mathbf{f}_I$ is at most $\delta + \epsilon_1 + \eta$. Since $\delta + \eta < 1/8$, choose ϵ_1 so that this value is still below 0.15 by a constant ϵ_2. Then Lemma 5 also gives $\mathbf{Pr}[R(\mathbf{v}^\star) > 0.15Q] \le N^{-\Omega(1)}$. A union bound over the two bad events completes the proof.

Furthermore, the independence between $\mathbf{f}$ and I naturally holds because the bounded channel only observes the codeword $\mathbf{c}$ and has no access to pk or sk. From the channel's perspective, the uniform random offset $\mathbf{z}$ effectively acts as a one-time pad that completely decorrelates $\mathbf{c}$ from the secret key, guaranteeing that the resulting error vector $\mathbf{f}$ is statistically independent of the hidden window I. Then, weak robustness against δ-bounded channel follows.

Theorem 6 (Weak soundness). *For every fixed* $\mathbf{u} \in \mathbb{F}_2^m$*, Construction 1 satisfies* $\mathbf{Pr}_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda)}[\mathsf{Decode}(sk, \mathbf{u}) = 1] \le \beta(\lambda)$*, where* $\beta(\lambda) \le 1/\operatorname{poly}(N)$*.*

Proof. The vector $(\mathbf{u} - \mathbf{z})_I$ is uniform over $\mathbb{F}_2^Q$, because $\mathbf{z}$ is uniform and included in the key generation randomness. For a fixed candidate $\mathbf{v}$, the center $\mathbf{M}_{I,J}\mathbf{v}$ is fixed after the key is fixed, so the probability that $(\mathbf{u} - \mathbf{z})_I$ lands within Hamming radius τ of it is at most $2^{-Q(1-\mathbf{H}(0.15))}$ by Lemma 1. The decoder checks 2^t candidates. Since $t = Q/3$ and $1 - \mathbf{H}(0.15) > 1/3$, the union bound gives $\beta(\lambda) \le 2^t \cdot 2^{-Q(1-\mathbf{H}(0.15))} = 2^{-\Omega(Q)} = 1/\operatorname{poly}(N)$.

Theorem 7 (Pseudorandomness). *If the* $\mathsf{DPLPN}(m, N, r, d, Q, \eta)$ *holds then Construction 1 is pseudorandom.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary making $L = \mathrm{poly}(\lambda)$ encoding queries. As established in Remark 2, the basic DPLPN assumption implies its multi-sample matrix variant. We rely on this matrix form to construct a reduction $\mathcal{B}$ that breaks the DPLPN assumption using $\mathcal{A}$.

The reduction $\mathcal{B}$ receives a matrix challenge pair $(\mathbf{A}, \mathbf{Y}) \in \mathbb{F}_2^{m\times r} \times \mathbb{F}_2^{m\times L}$, where $\mathbf{A} = \mathbf{MT}$ and $\mathbf{Y}$ is either $\mathbf{AS} + \mathbf{E}$ or $\mathbf{U}$. $\mathcal{B}$ samples $\mathbf{z} \leftarrow \mathbb{F}_2^m$, publishes $pk = (\mathbf{A}, \mathbf{z})$, and answers $\mathcal{A}$'s j-th query with $\mathbf{c}_j = \mathbf{Y}_{*,j} + \mathbf{z}$.

- If $\mathbf{Y} = \mathbf{AS}+\mathbf{E}$, then $\mathbf{c}_j = \mathbf{As}_j+\mathbf{e}_j+\mathbf{z}$, perfectly simulating the real encoding oracle $\mathsf{Encode}(pk)$.
- If $\mathbf{Y} = \mathbf{U}$, since $\mathbf{z}$ is fixed in pk and $\mathbf{U}_{*,j}$ is uniform and independent, $\mathbf{c}_j$ is perfectly uniform over $\mathbb{F}_2^m$, simulating the random oracle $\mathcal{U}$.

Thus, $\mathcal{A}$'s distinguishing advantage is exactly bounded by $\mathcal{B}$'s advantage against the matrix DPLPN assumption, which is negligible. Then, Construction 1 is pseudorandom.

Theorem 8 (PRC from Dense-Planted LPN). *Assume* $\mathsf{DPLPN}(m, N, r, d, Q, \eta)$ *holds in the canonical regime, and let* $\delta + \eta < 1/8$. *Construction 1 is a weak zero-bit public-key PRC with weak robustness* $\alpha(\lambda) \geq 1 - 1/\mathrm{poly}(N)$ *against the* δ*-bounded channel, weak soundness error* $\beta(\lambda) \leq 1/\mathrm{poly}(N)$*, and pseudorandomness.*

Furthermore, applying the generic amplification of Lemma 6 yields a zero-bit public-key PRC with standard robustness and soundness against the $(\delta - \epsilon)$*-bounded channel, for any fixed* $\epsilon > 0$*, while retaining pseudorandomness.*

4 Towards LWE-based Pseudorandom Code

This section explores the possibility of using the template of Section 3.2 to get a PRC based on LWE-type assumption. The pseudorandomness argument follows an analogous 'Dense-Planted LWE' assumption, whereas the decoder now needs an algorithm for a mixed-error decoding problem formalized below.

Write $\ell_q \stackrel{\mathsf{def}}{=} \lceil \log_2 q \rceil$ and let $\mathbf{G}_L := \mathbf{I}_L \otimes (1, 2, 4, \ldots, 2^{\ell_q - 1}) \in \mathbb{Z}_q^{L \times L\ell_q}$ denote the gadget matrix such that $\mathbf{b} \in \{0,1\}^{L\ell_q}$ is the bit decomposition of $\mathbf{G}_L\mathbf{b} \in \mathbb{Z}_q^L$; let $\mathbf{G}_L^{-1} : \mathbb{Z}_q^L \to \{0,1\}^{L\ell_q}$ be the bit decomposition function.

Definition 5 (Mixed error decoding). *The mixed error decoding problem* $\mathsf{MED}_{n,m,q,B,B'}$ *is the following promised search problem.*

- *Parameters: dimension* n*, length* m*, modulus* q*, norm bound* B*, and Hamming-weight bound* B'*, where* $m = \mathtt{poly}(n)$.
- *Input:* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ *and* $\mathbf{v} \in \mathbb{Z}_q^m$.
- *Promise: there exist* $\mathbf{x} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathbb{Z}_q^m$, *and* $\mathbf{e}' \in \{-1, 0, 1\}^{m\ell_q}$ *such that*

$$\|\mathbf{e}\|_\infty < B, \qquad |\mathbf{e}'| < B', \qquad \mathbf{v} = \mathbf{Ax} + \mathbf{e} + \mathbf{G}_m\mathbf{e}' \pmod q.$$

- *Output: a tuple* $(\mathbf{x}, \mathbf{e}, \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m \times \{-1, 0, 1\}^{m\ell_q}$ *witnessing the promise.*

4.1 Candidate PRC Construction

We now give the candidate construction. The construction is syntactically the same as the Dense-Planted LPN construction, except that the decoder requires an algorithm that solves the mixed error decoding problem, denoted $\mathsf{SolveMED}$.

We start by defining the Dense-Planted LWE assumption.

Definition 6 (Planted matrix distributions over $\mathbb{Z}_q$). *Let $m, N, d, Q \in \mathbb{N}$ be parameters and set $t := \lfloor Q/3 \rfloor$. Define two distributions over $\mathbb{Z}_q^{m \times N}$.*

- $\mathcal{D}_{\mathsf{rand}}^{m,N,d,q}$*: sample* $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{rand}}^{m,N,d,q}$*; then replace each non-zero entry of* $\mathbf{M}$ *by an independently, uniformly sampled element in* $\mathbb{Z}_q$*.*
- $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q,q}$*: sample* $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q,q}$ *along with hidden planted sets* (I, J)*; then replace each non-zero entry of* $\mathbf{M}$ *by an independently, uniformly sampled element in* $\mathbb{Z}_q$*.*

Assumption 2 (Dense-Planted LWE) *Let λ be the security parameter and let $m = m(\lambda), N = N(\lambda), r = r(\lambda), d = d(\lambda), Q = Q(\lambda)$, be polynomially bounded parameters. Let $q = q(\lambda)$ be a modulus, and let χ_λ be an efficiently sampleable error distribution over $\mathbb{Z}_q$. We say that the (m, N, r, d, Q, q, χ)-Dense-Planted LWE assumption, denoted $\mathsf{DPLWE}(m, N, r, d, Q, q, \chi)$, holds if for every PPT adversary $\mathcal{A}$,*

$$\left|\mathbf{Pr}\left[\mathcal{A}(\mathbf{MT}, \mathbf{MTs} + \mathbf{e}) = 1\right] - \mathbf{Pr}\left[\mathcal{A}(\mathbf{MT}, \mathbf{u}) = 1\right]\right| \leq \mathtt{negl}(\lambda),$$

where $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q,q}, \mathbf{T} \leftarrow \mathbb{Z}_q^{N \times r}, \mathbf{s} \leftarrow \mathbb{Z}_q^r, \mathbf{e} \leftarrow \chi^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$.

Construction 2 (Candidate zero-bit PRC from Dense-Planted LWE) *Fix parameters m, N, d, Q, r as in Remark 1. Moreover, choose a prime modulus $q = \lambda^6$ and error distribution $\mathcal{D}_{\sqrt{N}}$. Let $B_{\mathsf{in}} = N, B_{\mathsf{out}} = (\delta + \varepsilon) Q \ell_q$ where $\delta, \varepsilon \in (0, 1/10)$ and $\delta + \varepsilon \leq 1/10$. Define $\Pi_{\mathsf{DPLWE}} = (\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ over alphabet $\mathbb{F}_2$ as follows.*

$\mathsf{KeyGen}(1^\lambda)$**.** *Sample* $\mathbf{M} \leftarrow \mathcal{D}_{\mathsf{plant}}^{m,N,d,Q,q}$ *together with its hidden planted sets* (I, J)*. Sample* $\mathbf{T} \leftarrow \mathbb{Z}_q^{N \times r}$*, set* $\mathbf{A} := \mathbf{MT} \in \mathbb{Z}_q^{m \times r}$*, and sample* $\mathbf{z} \leftarrow \mathbb{Z}_q^m$*. Output* $pk = (\mathbf{A}, \mathbf{z})$*,* $sk = (I, J, \mathbf{M}_{I,J}, \mathbf{z})$*.*

$\mathsf{Encode}(pk)$**.** *Parse* $pk = (\mathbf{A}, \mathbf{z})$*. Sample* $\mathbf{s} \leftarrow \mathbb{Z}_q^r$ *and* $\mathbf{e} \leftarrow \mathcal{D}_{\sqrt{N}}^m$*. Compute* $\mathbf{c} = \mathbf{As} + \mathbf{e} + \mathbf{z} \pmod q$ *and output* $\mathbf{G}_m^{-1}(\mathbf{c})$*.*

$\mathsf{Decode}(sk, \widetilde{\mathbf{c}})$**.** *Parse* $sk = (I, J, \mathbf{C}, \mathbf{z})$*, where* $\mathbf{C} = \mathbf{M}_{I,J} \in \mathbb{Z}_q^{Q \times t}$*. Compute* $\mathbf{y} := (\mathbf{G}_m \widetilde{\mathbf{c}} - \mathbf{z})_I \in \mathbb{Z}_q^Q$*. Run* $(\mathbf{v}, \mathbf{e}_0, \mathbf{e}_1) \leftarrow \mathsf{SolveMED}_{t,Q,q,B_{\mathsf{in}},B_{\mathsf{out}}}(\mathbf{C}, \mathbf{y})$*. If the algorithm returns a tuple* $(\mathbf{v}, \mathbf{e}_0, \mathbf{e}_1)$ *satisfying*

$$\|\mathbf{e}_0\|_\infty < B_{\mathsf{in}}, \qquad |\mathbf{e}_1| < B_{\mathsf{out}}, \qquad \mathbf{y} = \mathbf{Cv} + \mathbf{e}_0 + \mathbf{G}_Q \mathbf{e}_1 \pmod q,$$

then output 1*. Otherwise output* $\perp$*.*

Decoding efficiency. It is easy to see that key generation and encoding run in polynomial time, while decoding runs in polynomial time plus the running time of $\mathsf{SolveMED}_{t,Q,q,B_{\mathsf{in}},B_{\mathsf{out}}}$. If $\mathsf{SolveMED}_{t,Q,q,B_{\mathsf{in}},B_{\mathsf{out}}}$ runs in time $2^{O(Q)} \cdot \mathtt{poly}(q)$, the decoding algorithm runs in polynomial time.

Theorem 9 (Conditional PRC from Dense-Planted LWE). *Let $\delta, \varepsilon \in (0, 1/10)$ be such that $\delta + \varepsilon \le 1/8$. Let N, m, Q, d, r be as in Remark 1 and*

$$\chi = \mathcal{D}_{\sqrt{N}}, \quad q = \lambda^6, \quad B_{\mathsf{in}} = \lambda, \quad B_{\mathsf{out}} = (\delta + \varepsilon) Q \log q.$$

Assume $\mathsf{DPLWE}(m, N, r, d, Q, q, \chi)$ holds and $\mathsf{MED}_{Q/3,Q,q,B_{\mathsf{in}},B_{\mathsf{out}}}$ can be solved in $2^{O(Q)} \cdot \mathtt{poly}(q) = \mathtt{poly}(N)$ time. Then Construction 2 is a zero-bit public-key PRC against δ-bounded channels.

It is worth noting that no security amplification is needed for the LWE-based (conditional) construction. The proof follows the same structure as the Dense-Planted LPN and is deferred to Section C.1.

4.2 On the Mixed Error Decoding Problem

When $B' = 0$, $\mathsf{MED}_{n,m,q,B,B'}$ is the usual bounded-distance decoding problem for the lattice generated by $\mathbf{A}$, and hence one may solve it via a closest-vector or bounded-distance decoding algorithm in $O(2^n)$, e.g., [SD15].

We note that the MED problem is closely related to the Learning with Two Errors assumption introduced by Ghosal, Jain, Lou, Sahai, and Vafa [GJL+25].

Definition 7 (Learning with Two Errors, informal [GJL+25]). *Let χ_{sm} be a small LWE error distribution over $\mathbb{Z}_q$, and let χ_{sp} be a sparse error distribution over $\mathbb{Z}_q$, which outputs zero with high probability and otherwise outputs a random element in $\mathbb{Z}_q$. The decisional LW2E assumption states that*

$$\left(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}^{(1)} + \mathbf{e}^{(2)}\right) \approx_c (\mathbf{A}, \mathbf{u}),$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e}^{(1)} \leftarrow \chi_{\mathsf{sm}}^m, \mathbf{e}^{(2)} \leftarrow \chi_{\mathsf{sp}}^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$.

The relation to MED is obtained by bit-decomposing $e^{(2)}$. Every $\mathbf{e}^{(2)} \in \mathbb{Z}_q^m$ has a binary representation $\mathbf{e}'' \in \{0,1\}^{m\ell_q}$ such that $\mathbf{e}^{(2)} = \mathbf{G}_m \mathbf{e}''$. If $\mathbf{e}^{(2)}$ has at most w nonzero coordinates, then $|\mathbf{e}''| \le w\ell_q$. Therefore, if $\left\|\mathbf{e}^{(1)}\right\|_\infty < B$ and $|\mathbf{e}^{(2)}| \le w$, then the search-LW2E sample $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}^{(1)} + \mathbf{e}^{(2)}$ is an instance of $\mathsf{MED}_{n,m,q,B,w\ell_q}$ with witness $(\mathbf{s}, \mathbf{e}^{(1)}, \mathbf{e}'')$.

Remark 3 (Why our MED solver does not contradict LW2E hardness). The decoder in Construction 2 only requires solving the MED instance of dimension n within time $O(2^{O(n)} \cdot \mathtt{poly}(q))$, which is exponential in n. Hence, the MED solver needed by the PRC template is compatible with the polynomial-time hardness of LW2E.

5 CCA Security, CCA1 Pseudorandomness, and Robustness

Let $\mathsf{PRC} = (\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ be a public-key pseudorandom code with message space $\mathcal{M}_\lambda$ and codeword space $\mathcal{C}_\lambda = \{0,1\}^{n(\lambda)}$. In a ***chosen codeword attack (CCA)***, the adversary has access to an encoding oracle as well as a decoding oracle.

5.1 The CCA Security Definition in [AAC+25]

We first recall the definition in [AAC+25]. For an ordered transcript $Q = ((m_1, c_1), \ldots, (m_t, c_t))$ of query-answer pairs of encoding oracle and a word $x \in \mathcal{C}_\lambda$, define $Q_\delta(x) \stackrel{\mathsf{def}}{=} \{i \in [t] : |c_i \oplus x| \le \delta n\}$. When $Q_\delta(x) \neq \emptyset$, define $\mathsf{Patch}_{Q,\delta}(x)$ to sample $i \leftarrow Q_\delta(x)$ uniformly and output m_i. This defines the behavior of the idealized decoding oracle when the query lies in the Hamming balls around the codewords that have already appeared in the encoding transcript.

Definition 8 (δ-CCA security for public-key PRCs [AAC+25]). *For $b \in \{0,1\}$, define the experiment $\mathcal{G}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, b)$ as follows.*

1. *The challenger samples $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, initializes $Q \leftarrow \emptyset$, and sends pk to $\mathcal{A}$.*
2. *The adversary may adaptively issue encoding and decoding queries.*
   - *On an encoding query $m \in \mathcal{M}_\lambda$, the challenger samples*

$$c \leftarrow \begin{cases} \mathsf{Encode}(pk, m), & b = 0, \\ \mathcal{C}_\lambda, & b = 1, \end{cases}$$

   where $c \leftarrow \mathcal{C}_\lambda$ denotes a uniformly random codeword. It appends (m, c) to Q and returns c.
   - *On a decoding query $x \in \mathcal{C}_\lambda$, the challenger returns*

$$\mathcal{O}^{b,Q}_{\mathsf{Decode}}(x) = \begin{cases} \mathsf{Decode}(sk, x), & b = 0, \\ \mathsf{Patch}_{Q,\delta}(x), & b = 1 \text{ and } Q_\delta(x) \neq \emptyset, \\ \mathsf{Decode}(sk, x), & b = 1 \text{ and } Q_\delta(x) = \emptyset. \end{cases}$$

3. *The adversary outputs a bit $\widehat{b}$.*

The CCA advantage of $\mathcal{A}$ is

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \stackrel{\mathsf{def}}{=} \left|\mathbf{Pr}\left[\mathcal{G}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, 0) = 1\right] - \mathbf{Pr}\left[\mathcal{G}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, 1) = 1\right]\right|.$$

We say that PRC satisfies δ-CCA security if $\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \le \mathtt{negl}(\lambda)$ for every efficient adversary $\mathcal{A}$.

The definition above captures robustness and pseudorandomness in a single game, which deviates from the convention that the correctness notion (i.e., robustness here) and the security notion are defined separately. Such a separation not only makes the definition more intuitive, but also makes proofs modular. Next, we show that the above definition can be equivalently expressed as two separate properties.

5.2 Decoupling CCA Security

Definition 9 (CCA-pseudorandomness). *For $b \in \{0,1\}$, define $\mathcal{G}^{\mathsf{pr\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, b)$ as follows.*

1. *The challenger samples $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, initializes $Q \leftarrow \emptyset$, and sends pk to $\mathcal{A}$.*
2. *The adversary may adaptively issue encoding and decoding queries.*
   - *On an encoding query $m \in \mathcal{M}_\lambda$, the challenger samples*

$$c \leftarrow \begin{cases} \mathsf{Encode}(pk, m), & b = 0, \\ \mathcal{C}_\lambda, & b = 1, \end{cases}$$

   appends (m, c) to Q, and returns c.
   - *On a decoding query $x \in \mathcal{C}_\lambda$, the challenger returns*

$$\mathcal{O}^Q_{\mathsf{Decode}}(x) = \begin{cases} \perp, & Q_\delta(x) \neq \emptyset, \\ \mathsf{Decode}(sk, x), & Q_\delta(x) = \emptyset. \end{cases}$$

3. *The adversary outputs a bit $\widehat{b}$.*

Define $\mathbf{Adv}^{\mathsf{pr\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \stackrel{\mathsf{def}}{=} \left|\mathbf{Pr}\left[\mathcal{G}^{\mathsf{pr\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, 0) = 1\right] - \mathbf{Pr}\left[\mathcal{G}^{\mathsf{pr\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, 1) = 1\right]\right|$. *We say that* PRC *is δ-CCA-pseudorandom if* $\mathbf{Adv}^{\mathsf{pr\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \leq \mathtt{negl}(\lambda)$ *for every efficient adversary $\mathcal{A}$.*

Definition 10 (Decisional adaptive robustness with decoder access). *For $b \in \{0,1\}$, define $\mathcal{G}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda, b)$ as follows.*

1. *The challenger samples $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, initializes $Q \leftarrow \emptyset$, and sends pk to $\mathcal{A}$.*
2. *The adversary may adaptively issue encoding and decoding queries.*
   - *On an encoding query $m \in \mathcal{M}_\lambda$, the challenger samples $c \leftarrow \mathsf{Encode}(pk, m)$, appends (m, c) to Q, and returns c.*
   - *On a decoding query $x \in \mathcal{C}_\lambda$, the challenger returns*

$$\mathcal{O}^{b,Q}_{\mathsf{Decode}}(x) = \begin{cases} \mathsf{Decode}(sk, x), & b = 0, \\ \mathsf{Patch}_{Q,\delta}(x), & b = 1 \text{ and } Q_\delta(x) \neq \emptyset, \\ \mathsf{Decode}(sk, x), & b = 1 \text{ and } Q_\delta(x) = \emptyset. \end{cases}$$

3. *The adversary outputs a bit* $\widehat{b}$.

Define $\mathbf{Adv}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \stackrel{\mathsf{def}}{=} \left|\mathbf{Pr}\left[\mathcal{G}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda,0)=1\right] - \mathbf{Pr}\left[\mathcal{G}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda,1)=1\right]\right|$. *We say that* PRC *satisfies decisional adaptive* δ*-robustness with decoder access if* $\mathbf{Adv}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \le \mathtt{negl}(\lambda)$ *for every efficient adversary* $\mathcal{A}$.

Lemma 8 (Decoupling the ideal CCA definition). *For every efficient adversary* $\mathcal{A}$ *attacking the CCA experiment of Definition 8, there exist efficient adversaries* $\mathcal{B}$ *and* $\mathcal{C}$ *with comparable running time such that*

$$\mathbf{Adv}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \le \mathbf{Adv}^{\mathsf{drob\text{-}cca}}_{\mathcal{B},\mathsf{PRC},\delta}(\lambda) + \mathbf{Adv}^{\mathsf{pr\text{-}cca}}_{\mathcal{C},\mathsf{PRC},\delta}(\lambda).$$

Consequently, δ*-CCA-pseudorandomness together with decisional adaptive* δ*-robustness with decoder access implies* δ*-CCA security in the sense of Definition 8.*

Proof. Let H_0 be the real CCA experiment $\mathcal{G}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda,0)$: encoding queries are answered by real codewords and decoding queries are answered by $\mathsf{Decode}(sk,\cdot)$. Let H_1 answer encoding queries by real codewords as in H_0, but replace the decoding oracle by the patched oracle that returns $\mathsf{Patch}_{Q,\delta}(x)$ whenever $Q_\delta(x) \neq \emptyset$ and returns $\mathsf{Decode}(sk,x)$ otherwise. Let H_2 be the random CCA experiment $\mathcal{G}^{\mathsf{cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda,1)$.

The gap between H_0 and H_1 is exactly the decisional robustness game of Definition 10: both hybrids use real encodings, and they only differ in whether decoding inside transcript Hamming balls is performed by the real decoder or by the patched decoder.

The gap between H_1 and H_2 is bounded by CCA-pseudorandomness. The reduction forwards encoding queries to its own encoding oracle and records the returned transcript Q. On a decoding query x, if $Q_\delta(x) \neq \emptyset$, it answers locally using $\mathsf{Patch}_{Q,\delta}(x)$; otherwise, it forwards x to its decoding oracle. Thus the forbidden-ball response of Definition 9 is never triggered. The simulation is H_1 when the pseudorandomness challenger uses real encodings and H_2 when it uses random encodings. The claim follows from the triangle inequality.

5.3 Search and Witness Formulations of Robustness

We note that there are different but natural ways to define robustness with decoder access. We present them below and discuss the relationships among different definitions.

Definition 11 ((Search) Adaptive robustness with decoder access). *Define the experiment* $\mathcal{G}^{\mathsf{srob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda)$ *as follows.*

1. *The challenger samples* $(pk,sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, *initializes* $Q \leftarrow \emptyset$, *and sends* pk *to* $\mathcal{A}$.
2. *The adversary may adaptively issue encoding and decoding queries.*
   - *On an encoding query* $m \in \mathcal{M}_\lambda$, *the challenger samples* $c \leftarrow \mathsf{Encode}(pk,m)$, *appends* (m,c) *to* Q, *and returns* c. *The index of this transcript entry is its position in* Q.

– *On a decoding query* $x \in \mathcal{C}_\lambda$*, the challenger returns* $\mathsf{Decode}(sk, x)$.

3. *The adversary outputs an index* i^* *and a word* $\widehat{c} \in \mathcal{C}_\lambda$.

The experiment outputs 1 *if* $i^* \in [|Q|]$ *and, writing* (m_{i^*}, c_{i^*}) *for the corresponding transcript entry,* $|c_{i^*} \oplus \widehat{c}| \le \delta n$ *and* $\mathsf{Decode}(sk, \widehat{c}) \neq m_{i^*}$. *Otherwise, it outputs* 0. *Define* $\mathbf{Adv}^{\mathsf{srob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \stackrel{\mathsf{def}}{=} \mathbf{Pr}\left[\mathcal{G}^{\mathsf{srob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda) = 1\right]$. *We say that* PRC *satisfies search adaptive* δ*-robustness with decoder access if* $\mathbf{Adv}^{\mathsf{srob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \le \mathtt{negl}(\lambda)$ *for every efficient adversary* $\mathcal{A}$.

Lemma 9. *Search adaptive* δ*-robustness with decoder access and decisional adaptive* δ*-robustness with decoder access are equivalent up to a polynomial loss.*

The proof is deferred to Section C.2.

Definition 12 (Witness adaptive robustness with decoder access). *Let* $\mathcal{R}_\lambda$ *denote the randomness space of* Encode. *Define the experiment* $\mathcal{G}^{\mathsf{wrob\text{-}dec}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda)$ *as follows.*

1. *The challenger samples* $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$.
2. *The adversary adaptively query the decoding oracle* $\mathcal{O}_{\mathsf{Decode}}(x) = \mathsf{Decode}(sk, x)$.
3. *The adversary outputs a tuple* $(m, r, \widehat{c})$ *with* $m \in \mathcal{M}_\lambda$, $r \in \mathcal{R}_\lambda$, *and* $\widehat{c} \in \mathcal{C}_\lambda$.

The experiment outputs 1 *if, for* $c := \mathsf{Encode}(pk, m; r)$,

$$|c \oplus \widehat{c}| \le \delta n \quad \textit{and} \quad \mathsf{Decode}(sk, \widehat{c}) \neq m.$$

Otherwise it outputs 0. *Define* $\mathbf{Adv}^{\mathsf{wrob\text{-}dec}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \stackrel{\mathsf{def}}{=} \mathbf{Pr}\left[\mathcal{G}^{\mathsf{wrob\text{-}dec}}_{\mathcal{A},\mathsf{PRC},\delta}(1^\lambda) = 1\right]$. *We say that* PRC *satisfies witness adaptive* δ*-robustness with decoder access if the advantage function* $\mathbf{Adv}^{\mathsf{wrob\text{-}dec}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \le \mathtt{negl}(\lambda)$ *for every efficient adversary* $\mathcal{A}$.

Remark 4 (Relationship with the earlier adaptive-robustness definition). In Definition 11, the erroneous codeword is sampled by the challenger. Definition 12 allows the adversary to choose the encoding randomness and therefore catches rare bad randomness values. It is easy to see definition 12 implies definition 11 by simulating the encoding oracle as follows: The reduction samples the encoding randomness itself, returns $c = \mathsf{Encode}(pk, m; r)$, and records (m, r, c); a transcript witness $(i, \widehat{c})$ then gives the randomness witness $(m_i, r_i, \widehat{c})$. Moreover, Definition 12 implies the previous adaptive robustness (without decoder access) by ignoring the decoding oracle.

5.4 CCA1-pseudorandomness

Analogous to public-key encryption, we consider security under ***pre-challenge*** CCA attack, referred to as CCA1 security. In this model, the adversary can only query the decoding oracle before the challenge phase.

Definition 13 (CCA1 pseudorandomness). *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an efficient two-stage adversary. For* $b \in \{0, 1\}$, *define* $\mathcal{G}^{\mathsf{pr\text{-}cca1}}_{\mathcal{A},\mathsf{PRC}}(1^\lambda, b)$ *as follows.*

1. *The challenger samples* $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}_1$.
2. *The first-stage adversary* $\mathcal{A}_1$ *may adaptively query the decoding oracle* $\mathcal{O}_{\mathsf{Decode}}(x) = \mathsf{Decode}(sk, x)$. *It then outputs a state* st *and a challenge message* $m^* \in \mathcal{M}_\lambda$.
3. *The challenger samples* $c^* \leftarrow \begin{cases} \mathsf{Encode}(pk, m^*), & b = 0, \\ \mathcal{C}_\lambda, & b = 1. \end{cases}$
4. *The second-stage adversary* $\mathcal{A}_2(pk, \mathsf{st}, c^*)$ *outputs a bit* $\widehat{b}$. *It receives no decoding oracle.*

Define $\mathbf{Adv}^{\mathsf{pr\text{-}cca1}}_{\mathcal{A},\mathsf{PRC}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\mathbf{Pr}\left[\mathcal{G}^{\mathsf{pr\text{-}cca1}}_{\mathcal{A},\mathsf{PRC}}(1^\lambda, 0) = 1\right] - \mathbf{Pr}\left[\mathcal{G}^{\mathsf{pr\text{-}cca1}}_{\mathcal{A},\mathsf{PRC}}(1^\lambda, 1) = 1\right]\right|$. *We say that* PRC *is CCA1-pseudorandom if* $\mathbf{Adv}^{\mathsf{pr\text{-}cca1}}_{\mathcal{A},\mathsf{PRC}}(\lambda) \leq \mathtt{negl}(\lambda)$ *for every PPT adversary* $\mathcal{A}$.

6 Realizing CCA1-Secure PRC

This section presents our generic construction of a CCA1-pseudorandom PRC from a CPA-secure PRC with the following special property and a robust hinting PRG.

Definition 14 (Recovery from randomness). *Let* Π *be a PRC with* δ*-adaptive robustness and codeword length* n. *We say* Π *allows* ***recovery from randomness*** *if there exists a deterministic algorithm* Rec *such that for all* $\lambda \in \mathbb{N}$, *all messages* m, *and all* $(pk, sk) \leftarrow \Pi.\mathsf{KeyGen}(1^\lambda)$, *it holds that*

$$\mathop{\mathbf{Pr}}_{r \leftarrow \mathcal{R}}\left[\forall e \in \{0,1\}^n \text{ s.t. } |e| \leq \delta n \implies \mathsf{Rec}(pk, \Pi.\mathsf{Encode}(pk, m; r) \oplus e, r) = m\right] = 1.$$

Moreover, we may assume that $\mathsf{Rec}(pk, \widehat{c}, r)$ *returns either a message* $y \in \{0,1\}^{\ell_{msg}}$ *or* $\perp$ *and* $\mathsf{Rec}(pk, \widehat{c}, r) = y \neq \perp \implies |\widehat{c} \oplus \mathsf{Encode}(pk, y; r)| \leq \delta n$.

Hinting PRG can be instantiated under any of the CDH/LWE assumption (Theorem 4), and hence we have the following theorem.

Theorem 10. *Assume the CDH/LWE assumption holds. Suppose that there exists a CPA-secure PRC that encodes* $\mathtt{poly}(\lambda)$*-bit messages, allows recovery from randomness, and is adaptively* 0.1*-robust. Then there exists a PRC that encodes* λ*-bit messages, is CCA1-pseudorandom, and satisfies search adaptive* 0.0005*-robustness with decoder access.*

The PRC construction by Christ and Gunn, which is based on the subexponential LPN assumption [CG24, AAC+25]. In Section C.3, we show that it can be transformed into a PRC that allows recovery from randomness and encodes multi-bit messages. Combining this with Theorem 10 proves Theorem 3.

The rest of this section is organized as follows.

– Section 6.1 introduces a primitive called ***robust tag-based equivocal bit commitment (robust TBEQC)***, which can be based on one-way functions and plays an important role in our construction. Its construction based on one-way functions is given in Section B.

– Section 6.2 presents our PRC construction. We defer the detailed proofs of its robustness with decoder access and CCA1-pseudorandomness to Section 7.2 and Section 7.3, respectively.

6.1 Robust Tag-Based Equivocal Bit Commitment

Definition 15. *A $(\delta_{com}, \delta_{tag}, \gamma, n_{tag}, n_{com}, \ell_{open})$-robust tag-based equivocal bit commitment (robust TBEQC) scheme consists of four PPT algorithms* $(\mathsf{CKG}, \mathsf{Com}, \mathsf{Verify}, \mathsf{EQSetup})$ *with the following syntax.*

- $\mathsf{CKG}(1^\lambda) \mapsto ck$*: outputs a commitment key* ck*.*
- $\mathsf{Com}(ck, t, b; \rho) \mapsto \mathsf{com} \in \{0,1\}^{n_{com}}$*: takes a commitment key* ck*, a tag* $t \in \{0,1\}^{n_{tag}}$*, a message bit* $b \in \{0,1\}$*, and randomness* $\rho \in \{0,1\}^{\ell_{open}}$*; it outputs a commitment* $\mathsf{com} \in \{0,1\}^{n_{com}}$*.*
- $\mathsf{Verify}(ck, \widehat{t}, \widehat{\mathsf{com}}, \rho, b) \mapsto \{0,1\}$*: takes a commitment key* ck*, a possibly corrupted tag* $\widehat{t}$*, a possibly corrupted commitment* $\widehat{\mathsf{com}}$*, an opening* $\rho \in \{0,1\}^{\ell_{open}}$*, and a bit* $b \in \{0,1\}$*; it outputs* 1 *to indicate that the commitment opens to* b*, and outputs* 0 *otherwise.*
- $\mathsf{EQSetup}(1^\lambda, t^*) \mapsto (\widetilde{ck}, \widetilde{\mathsf{com}}, \widetilde{\rho}_0, \widetilde{\rho}_1)$*: takes a special equivocal tag* $t^* \in \{0,1\}^{n_{tag}}$ *and outputs an equivocal key* $\widetilde{ck}$*, an equivocal commitment* $\widetilde{\mathsf{com}}$*, and two openings* $\widetilde{\rho}_0, \widetilde{\rho}_1 \in \{0,1\}^{\ell_{open}}$*.*

$(\delta_{com}, \delta_{tag})$-robustness. There exists a negligible function ν such that for all $\lambda \in \mathbb{N}$, $t \in \{0,1\}^{n_{tag}}$, and $b \in \{0,1\}$,

$$\mathbf{Pr}\left[\begin{array}{l}\forall e \in \{0,1\}^{n_{com}}, e' \in \{0,1\}^{n_{tag}}, \\ |e| \le \delta_{com} n_{com} \wedge |e'| \le \delta_{tag} n_{tag} \implies \mathsf{Verify}(ck, t \oplus e', \mathsf{com} \oplus e, \rho, b) = 1\end{array}\right] \ge 1 - \nu(\lambda),$$

where $ck \leftarrow \mathsf{CKG}(1^\lambda)$, $\rho \leftarrow \{0,1\}^{\ell_{open}}$, and $\mathsf{com} := \mathsf{Com}(ck, t, b; \rho)$.

Security properties. The following security properties are required.

1. ***Mode indistinguishability.*** *For all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{C},\mathcal{A}}(\lambda) := \left|\mathbf{Pr}[\mathsf{Expt}^{\mathsf{normal}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1] - \mathbf{Pr}[\mathsf{Expt}^{\mathsf{eq}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1]\right|$ is negligible, where $\mathsf{Expt}^{\mathsf{normal}}$ and $\mathsf{Expt}^{\mathsf{eq}}$ are defined as follows.*

$\mathsf{Expt}^{\mathsf{normal}}_{\mathcal{C},\mathcal{A}}(\lambda)$:	$\mathsf{Expt}^{\mathsf{eq}}_{\mathcal{C},\mathcal{A}}(\lambda)$:
$(t^*, b, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$	$(t^*, b, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\lambda)$
$ck \leftarrow \mathsf{CKG}(1^\lambda)$	$(\widetilde{ck}, \widetilde{\mathsf{com}}, \widetilde{\rho}_0, \widetilde{\rho}_1) \leftarrow \mathsf{EQSetup}(1^\lambda, t^*)$
$\rho \leftarrow \{0,1\}^{\ell_{open}}$	
$\mathsf{com} \leftarrow \mathsf{Com}(ck, t^*, b; \rho)$	
$b' \leftarrow \mathcal{A}_2(ck, \mathsf{com}, \rho, \mathsf{st})$	$b' \leftarrow \mathcal{A}_2(\widetilde{ck}, \widetilde{\mathsf{com}}, \widetilde{\rho}_b, \mathsf{st})$
Return b'.	*Return* b'.

2. ***Statistical binding in normal mode.*** *In the normal mode, no tag can be equivocal. Formally, it is required that the following quantity is negligible in* λ*:*

$$\Pr_{ck \leftarrow \mathsf{CKG}(1^\lambda)} \left[\begin{array}{c} \exists \widehat{t} \in \{0,1\}^{n_{tag}}, \rho_0, \rho_1 \in \{0,1\}^{\ell_{open}}, \widehat{\mathsf{com}} \in \{0,1\}^{n_{com}} \ \ s.t. \\ \wedge\ \mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_0, 0) = 1 \\ \wedge\ \mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_1, 1) = 1 \end{array} \right].$$

3. γ***-robust statistical binding in EQ mode.*** *In the EQ mode, only tags close to* t^* *or* $\overline{t^*}$ *can be equivocal. Formally, it is required that the following quantity is negligible in* λ*:*

$$\max_{t^* \in \{0,1\}^{n_{tag}}} \Pr \left[\begin{array}{c} \exists \widehat{t} \in \{0,1\}^{n_{tag}}, \rho_0, \rho_1 \in \{0,1\}^{\ell_{open}}, \widehat{\mathsf{com}} \in \{0,1\}^{n_{com}} \ \ s.t. \\ \gamma n_{tag} \le |\widehat{t} \oplus t^*| \le (1-\gamma) n_{tag} \\ \wedge\ \mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_0, 0) = 1 \\ \wedge\ \mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_1, 1) = 1 \end{array} \right],$$

where the probability is over $(\widetilde{ck}, \widetilde{\mathsf{com}}, \widetilde{\rho}_0, \widetilde{\rho}_1) \leftarrow \mathsf{EQSetup}(1^\lambda, t^*)$.

4. ***Pseudorandomness of commitments.*** *In normal mode, generated commitments are computationally indistinguishable from uniformly random strings of the same length. For all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the advantage*

$$\mathsf{Adv}^{\mathsf{pr}}_{\mathcal{C},\mathcal{A}}(\lambda) := \left| \Pr[\mathsf{Expt}^{\mathsf{pr\text{-}real}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Expt}^{\mathsf{pr\text{-}rand}}_{\mathcal{C},\mathcal{A}}(\lambda) = 1] \right|$$

is negligible, where $\mathsf{Expt}^{\mathsf{pr\text{-}real}}$ *and* $\mathsf{Expt}^{\mathsf{pr\text{-}rand}}$ *are defined as follows:*

$\mathsf{Expt}^{\mathsf{pr\text{-}real}}_{\mathcal{C},\mathcal{A}}(\lambda)$:	$\mathsf{Expt}^{\mathsf{pr\text{-}rand}}_{\mathcal{C},\mathcal{A}}(\lambda)$:
$ck \leftarrow \mathsf{CKG}(1^\lambda)$	$ck \leftarrow \mathsf{CKG}(1^\lambda)$
$(t, b, \mathsf{st}) \leftarrow \mathcal{A}_1(ck)$	$(t, b, \mathsf{st}) \leftarrow \mathcal{A}_1(ck)$
$\rho \leftarrow \{0,1\}^{\ell_{open}}$	
$\mathsf{com} \leftarrow \mathsf{Com}(ck, t, b; \rho)$	$u \leftarrow \{0,1\}^{n_{com}}$
$b' \leftarrow \mathcal{A}_2(\mathsf{com}, \mathsf{st})$	$b' \leftarrow \mathcal{A}_2(u, \mathsf{st})$
Return b'.	*Return* b'.

By mode indistinguishability, the pseudorandomness property also holds in EQ mode.

Theorem 11. *Let* $n_{tag}, n_{com}, \ell_{open} \in \mathtt{poly}(\lambda)$ *be three length functions and* $\delta_{com}, \delta_{tag}, \gamma, \epsilon \in (0, 1/2)$ *be constants. Suppose that*

- $\log \frac{1}{1-\gamma} > 2\,\mathbf{H}(\delta_{com} + \delta_{tag} + \epsilon)$*, and*
- $n_{com} \ge \max\left\{ \frac{n_{tag} + \ell_{open} + \lambda}{\zeta}, \quad \frac{\ln 2 \cdot \mathbf{H}(\delta_{tag}) \cdot n_{tag}}{\epsilon} \right\}$ *where* $\zeta = \log \frac{1}{1-\gamma} - 2\,\mathbf{H}(\delta_{com} + \delta_{tag} + \epsilon)$.

Then there exists a $(\delta_{com}, \delta_{tag}, \gamma, n_{tag}, n_{com}, \ell_{open})$*-robust TBEQC provided that one-way function exists.*

6.2 Construction

We construct a PRC $\Pi = (\mathsf{KeyGen}, \mathsf{Encode}, \mathsf{Decode})$ for messages $m \in \{0,1\}^k$. Our construction relies on the following ingredients.

1. **CPA-secure PRC with adaptive δ_{cpa}-robustness.** Let
$$\mathcal{E} = (\mathsf{CPA.KeyGen}, \mathsf{CPA.Encode}, \mathsf{CPA.Decode}, \mathsf{CPA.Rec})$$
be a public-key pseudorandom code encoding ℓ_{msg}-bit messages into n_{cpa}-bit codewords using ℓ_{rnd}-bit randomness. We assume $\mathcal{E}$ allows recovery from randomness.
2. **Robust TBEQC.**
Let $\mathcal{C} = (\mathsf{CKG}, \mathsf{Com}, \mathsf{Verify}, \mathsf{EQSetup})$ be a $(\delta_{com}, \delta_{tag}, \gamma, n_{tag}, n_{com}, \ell_{msg}-1)$-robust TBEQC. The commitment has opening length $\ell_{open} = \ell_{msg} - 1$.
3. **Error correcting codes.** Let $\mathcal{C}_{ecc} = (\mathsf{ECC.Enc}, \mathsf{ECC.Dec})$ be an error-correcting code capable of correcting up to αN errors, encoding κ-bit messages into N-bit codewords. Let $\mathcal{C}'_{ecc} = (\mathsf{ECC'.Enc}, \mathsf{ECC'.Dec})$ be an error-correcting code capable of correcting up to δ_{core} fraction of errors, encoding k-bit messages into n_{core}-bit codewords.
4. **Robust Hinting PRG.**
Let $\mathcal{H} = (\mathsf{HPRG.Setup}, \mathsf{HPRG.Eval})$ be a $(\kappa, N, n_{core}, \ell_{rnd})$-robust HPRG.

Parameter setting. Set $\kappa := 3\lambda$, $\alpha := 0.2$, $\delta_{cpa} = 0.1$. We choose N such that there exists an ECC that encodes a κ-bit string into an N-bit codeword and corrects up to an α-fraction of errors. Set

$$\delta_{com} := 0.005,\ \delta_{tag} := 0.030,\ \epsilon := 0.025,\ \gamma := 0.48,\ n_{tag} := 3n_{cpa},\ n_{com} := 18n_{cpa}.$$

One can verify that $(\delta_{com}, \delta_{tag}, \gamma, \epsilon, n_{com}, n_{tag}, \ell_{msg}-1)$ satisfies the condition of Theorem 11, so there exists a $(\delta_{com}, \delta_{tag}, \gamma, n_{tag}, n_{com}, \ell_{msg}-1)$-robust TBEQC by Theorem 11. [5] Set

$$\delta := 0.0005,\ n_{core} := N \cdot n_{cpa},\ \delta_{core} := 25\delta.$$

Choose $\mathsf{ECC'}$ that encodes k-bit messages into n_{core}-bit codewords and corrects up to a δ_{core}-fraction of errors. Such an ECC exists for $k \le \lambda^2$. An $(\kappa, N, n_{core}, \ell_{rnd})$-robust hinting PRG can be constructed assuming LWE/CDH according to Theorem 4.

Key generation $\mathsf{KeyGen}(1^\lambda)$*:*

1. Generate parameters for the Hinting PRG: $pp \leftarrow \mathsf{HPRG.Setup}(1^\lambda)$.
2. Generate $2N$ independent key pairs for the CPA-secure PRC: for $b \in \{0,1\}$ and $i \in [N]$, compute $(pk_{b,i}, sk_{b,i}) \leftarrow \mathsf{CPA.KeyGen}(1^\lambda)$.
3. Generate commitment keys $ck_i \leftarrow \mathsf{CKG}(1^\lambda)$ for each $i \in [N]$.
4. Output the public key $PK = (pp, \{ck_i\}_{i\in[N]}, \{pk_{b,i}\}_{b\in\{0,1\},i\in[N]})$ and the secret key $SK = (PK, \{sk_{b,i}\}_{b\in\{0,1\},i\in[N]})$.

[5] $\log \frac{1}{1-\gamma} = 0.943...$, $\mathbf{H}(\delta_{tag}) = 0.194...$, $\mathbf{H}(\delta_{com} + \delta_{tag} + \epsilon) = \mathbf{H}(0.06) = 0.327...$, $\ln 2 = 0.69...$.

Encoding $\mathsf{Encode}(PK, m)$*:*

1. Sample a random string $\sigma \leftarrow \{0,1\}^{\kappa}$ and encode it to generate the HPRG seed: $s = \mathsf{ECC.Enc}(\sigma) \in \{0,1\}^{N}$.
2. Compute the core mask and masked ciphertext: $K = \mathsf{HPRG.Eval}(pp, s, 0)$ and set $\mathsf{ct}_0 := K \oplus \mathsf{ECC'.Enc}(m)$.
3. Sample random tags $t_1, \ldots, t_N \leftarrow \{0,1\}^{n_{tag}}$.
4. For each $i \in [N]$:
   (a) Compute the evaluation randomness $\widetilde{r}_i = \mathsf{HPRG.Eval}(pp, s, i)$ and choose independent fresh randomness $r_i \leftarrow \{0,1\}^{\ell_{rnd}}$.
   (b) If $s_i = 0$: sample $\rho_i \leftarrow \{0,1\}^{\ell_{open}}$, compute $\mathsf{com}_i = \mathsf{Com}(ck_i, t_i, 0; \rho_i)$, and set $c_{0,i} = \mathsf{CPA.Encode}(pk_{0,i}, 1\|\rho_i; \widetilde{r}_i)$, $c_{1,i} = \mathsf{CPA.Encode}(pk_{1,i}, 0^{\ell_{msg}}; r_i)$.
   (c) If $s_i = 1$: sample $\rho_i \leftarrow \{0,1\}^{\ell_{open}}$, compute $\mathsf{com}_i = \mathsf{Com}(ck_i, t_i, 1; \rho_i)$, and set $c_{0,i} = \mathsf{CPA.Encode}(pk_{0,i}, 0^{\ell_{msg}}; r_i)$, $c_{1,i} = \mathsf{CPA.Encode}(pk_{1,i}, 1\|\rho_i; \widetilde{r}_i)$.
   (d) Set $\mathsf{ct}_i := t_i \| \mathsf{com}_i \| c_{0,i} \| c_{1,i}$.
5. Output the concatenated codeword $\mathsf{ct} = \mathsf{ct}_0 \| \mathsf{ct}_1 \| \cdots \| \mathsf{ct}_N$.

Decoding $\mathsf{Decode}(SK, \widehat{\mathsf{ct}})$*:* The decoding algorithm uses the subroutines Find and Check, defined in Figure 1 and Figure 2. Note that the Find routine only uses half of the CPA secret keys, namely $\{sk_{0,i}\}_{i \in [N]}$.

1. Parse $SK = (PK, \{sk_{b,i}\}_{b \in \{0,1\}, i \in [N]})$.
2. Compute $d = \mathsf{Find}(PK, SK, \widehat{\mathsf{ct}})$.
3. Set $\sigma := \mathsf{ECC.Dec}(d)$. If $\sigma = \perp$, return $\perp$.
4. Reconstruct the seed $s = \mathsf{ECC.Enc}(\sigma)$.
5. Parse $\widehat{\mathsf{ct}}$ to extract the first ciphertext component $\widehat{\mathsf{ct}}_0$. Recover the message candidate $m = \mathsf{ECC'.Dec}\left(\widehat{\mathsf{ct}}_0 \oplus \mathsf{HPRG.Eval}(pp, s, 0)\right)$.
6. Initialize $S = \emptyset$. For $i = 1, \ldots, N$, if $s_i = d_i$ and $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True}$, add i to S.
7. If $|S| \geq (1 - \alpha)N$, return m; otherwise, return $\perp$.

In Section 7, we show the above construction satisfies CCA1-psedorandomness and search adaptive 0.0005-robustness with decoder access, establishing Theorem 10.

Routine $\mathsf{Find}(PK, SK, \widehat{\mathsf{ct}})$

Inputs:

- Public key $PK = (pp, \{ck_i\}_{i\in[N]}, \{pk_{b,i}\}_{b\in\{0,1\},i\in[N]})$.
- Secret key $SK = (PK, \{sk_{b,i}\}_{b\in\{0,1\},i\in[N]})$. Only $sk_{0,i}$ is used.
- Codeword $\widehat{\mathsf{ct}} = \widehat{\mathsf{ct}}_0\| \cdots \|\widehat{\mathsf{ct}}_N$, where $\widehat{\mathsf{ct}}_i = \widehat{t}_i\|\widehat{\mathsf{com}}_i\|\widehat{c}_{0,i}\|\widehat{c}_{1,i}$ for $i \in [N]$.

Output: $d \in \{0,1\}^N$.

For each $i \in [N]$, do the following:

1. Let $y = \mathsf{CPA.Decode}(sk_{0,i}, \widehat{c}_{0,i})$.
2. If $y = 1\|\rho$ and $\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho, 0) = 1$, set $d_i = 0$. Otherwise, set $d_i = 1$.

Output $d = d_1 d_2 \dots d_N$.

Fig. 1. Routine Find using the 0-branch to recover the HPRG seed.

Routine $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i)$

Inputs: codeword $\widehat{\mathsf{ct}}$, HPRG seed s, and index $i \in [N]$.
Hardwired: public key $PK = (pp, \{ck_i\}_{i\in[N]}, \{pk_{b,i}\}_{b\in\{0,1\},i\in[N]})$.
Operations:

1. Parse $\widehat{\mathsf{ct}} = \widehat{\mathsf{ct}}_0\|\widehat{\mathsf{ct}}_1\| \cdots \|\widehat{\mathsf{ct}}_N$ where $\widehat{\mathsf{ct}}_i = \widehat{t}_i\|\widehat{\mathsf{com}}_i\|\widehat{c}_{0,i}\|\widehat{c}_{1,i}$.
2. Let $b := s_i$ and compute $r := \mathsf{HPRG.Eval}(pp, s, i)$.
3. Let $y := \mathsf{CPA.Rec}(pk_{b,i}, \widehat{c}_{b,i}, r)$. If $y = \perp$ or the first bit of y is 0, return False.
4. Parse $y = 1\|\rho$. By the assumed property of $\mathsf{CPA.Rec}$, whenever this point is reached, we have

$$|\widehat{c}_{b,i} \oplus \mathsf{CPA.Encode}(pk_{b,i}, 1\|\rho; r)| \le \delta_{cpa} n_{cpa}.$$

   This fact will be used in the CCA1 proof via the adaptive robustness of the underlying CPA-secure PRC.
5. If $\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho, b) = 0$, return False.
6. Return True.

Fig. 2. Routine Check to enforce signal validity.

7 Security Proofs for the CCA1-Secure PRC

In this section, we show that the construction in Section 6.2 satisfies CCA1-psedorandomness and search adaptive 0.0005-robustness with decoder access, establishing Theorem 10.

7.1 Shared Notation for the Security Proofs

The robustness and CCA1-pseudorandomness proofs both use variants of the same decoding routine. We collect the common notation here.

Branch-selecting finders and decoders. For a branch selector $z \in \{0,1\}^N$, define $\mathsf{Find}^z(PK, SK, \widehat{\mathsf{ct}})$ as follows. Parse

$$\widehat{\mathsf{ct}} = \widehat{\mathsf{ct}}_0 \| \widehat{\mathsf{ct}}_1 \| \cdots \| \widehat{\mathsf{ct}}_N, \qquad \widehat{\mathsf{ct}}_i = \widehat{t}_i \| \widehat{\mathsf{com}}_i \| \widehat{c}_{0,i} \| \widehat{c}_{1,i}.$$

For each $i \in [N]$, compute

$$y_i := \mathsf{CPA.Decode}(sk_{z_i,i}, \widehat{c}_{z_i,i}).$$

If $y_i = 1 \| \rho_i$ and

$$\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho_i, z_i) = 1,$$

then set $d_i := z_i$; otherwise set $d_i := 1 - z_i$. Output $d = d_1 \cdots d_N$. Let Decode^z be the top-level decoder obtained from Decode by replacing Find with Find^z and leaving the rest of the decoding algorithm unchanged. The implemented decoder is

$$\mathsf{Decode} = \mathsf{Decode}^{\mathbf{0}},$$

where $\mathbf{0} \in \{0,1\}^N$ is the all-zero selector. We also write $\mathbf{1} \in \{0,1\}^N$ for the all-one selector.

Certified openings. For $b \in \{0,1\}$, randomness $r \in \{0,1\}^{\ell_{rnd}}$, index $i \in [N]$, and a possibly malformed codeword $\widehat{\mathsf{ct}}$, define $\mathsf{OpenOK}_b(PK, \widehat{\mathsf{ct}}, i, r)$ as follows.

1. Parse $\widehat{\mathsf{ct}}_i = \widehat{t}_i \| \widehat{\mathsf{com}}_i \| \widehat{c}_{0,i} \| \widehat{c}_{1,i}$.
2. Compute $y := \mathsf{CPA.Rec}(pk_{b,i}, \widehat{c}_{b,i}, r)$.
3. If $y \neq 1 \| \rho$ for every ρ, output $\perp$. Otherwise, parse $y = 1 \| \rho$ and output y if

$$\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho, b) = 1.$$

   If verification rejects, output $\perp$.

Equivalently,

$$\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True} \iff \mathsf{OpenOK}_{s_i}\left(PK, \widehat{\mathsf{ct}}, i, \mathsf{HPRG.Eval}(pp, s, i)\right) \neq \perp.$$

Moreover, by certifying randomness recovery, whenever

$$\mathsf{OpenOK}_b(PK, \widehat{\mathsf{ct}}, i, r) = y \neq \perp,$$

we have

$$|\widehat{c}_{b,i} \oplus \mathsf{CPA.Encode}(pk_{b,i}, y; r)| \leq \delta_{cpa} n_{cpa}.$$

Thus a non-$\perp$ output of OpenOK is a certificate that the corresponding CPA component is close to a known codeword.

Validity and double openings. For a word $\widehat{\mathsf{ct}}$, say that branch b is ***valid*** at index i if

$$\mathsf{CPA.Decode}(sk_{b,i}, \widehat{c}_{b,i}) = 1\|\rho \quad \text{and} \quad \mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho, b) = 1$$

for some ρ. A ***double opening*** occurs at $(\widehat{\mathsf{ct}}, i)$ if both branches 0 and 1 are valid at index i.

7.2 Robustness

Let $L = n_{core} + N \cdot (n_{com} + n_{tag} + 2n_{cpa}) = 24N \cdot n_{cpa}$ be the length of the codewords of Π.

We prove robustness with decoder access in the sense of Definition 11. The underlying CPA-secure PRC is only assumed to satisfy the ordinary witness-based adaptive robustness notion, without decoder access. The proof uses $\mathsf{Decode}^{\mathbf{1}}$ only as an analysis decoder. The adversary's actual decoding oracle is always the implemented decoder $\mathsf{Decode}^{\mathbf{0}}$.

Theorem 12 (Robustness with decoder access). *If the parameters satisfy*

1. $\delta L \le \delta_{core} n_{core}$,
2. $\delta L \le \alpha N \cdot \min(\delta_{tag} n_{tag}, \delta_{com} n_{com}, \delta_{cpa} n_{cpa})$,

then Π satisfies search adaptive δ-robustness with decoder access.

Proof. Let $\mathcal{A}$ be a PPT adversary in the search adaptive robustness experiment with decoder access. The challenger answers encoding queries honestly and answers decoding queries using the implemented decoder $\mathsf{Decode}^{\mathbf{0}}$. Let

$$Q = ((m^{(1)}, \mathsf{ct}^{(1)}), \ldots, (m^{(q_E)}, \mathsf{ct}^{(q_E)}))$$

be the encoding transcript. For the j-th transcript codeword, write

$$\mathsf{ct}^{(j)} = \mathsf{ct}_0^{(j)} \| \mathsf{ct}_1^{(j)} \| \cdots \| \mathsf{ct}_N^{(j)}, \qquad \mathsf{ct}_i^{(j)} = t_i^{(j)} \| \mathsf{com}_i^{(j)} \| c_{0,i}^{(j)} \| c_{1,i}^{(j)}.$$

Let $s^{(j)} = \mathsf{ECC.Enc}(\sigma^{(j)})$ be the HPRG seed used in the j-th encoding, and let $\rho_i^{(j)}$ be the commitment opening at index i.

For $b \in \{0,1\}$, define the CPA message and randomness used in branch (b,i) of $\mathsf{ct}^{(j)}$ by

$$(Y_{b,i}^{(j)}, R_{b,i}^{(j)}) := \begin{cases} \left(1\|\rho_i^{(j)}, \mathsf{HPRG.Eval}(pp, s^{(j)}, i)\right), & b = s_i^{(j)}, \\ \left(0^{\ell_{msg}}, r_i^{(j)}\right), & b = 1 - s_i^{(j)}, \end{cases}$$

where $r_i^{(j)}$ is the fresh randomness used for the inactive branch. Thus

$$c_{b,i}^{(j)} = \mathsf{CPA.Encode}\left(pk_{b,i}, Y_{b,i}^{(j)}; R_{b,i}^{(j)}\right).$$

We condition on the standard correctness events. By the $(\delta_{com}, \delta_{tag})$-robustness of the TBEQC and a union bound over the polynomially many transcript commitments, except with negligible probability every transcript commitment continues to verify under its original opening after any local corruption within radii $\delta_{tag} n_{tag}$ and $\delta_{com} n_{com}$. By certifying randomness recovery and a union bound over the $2N$ CPA keys, except with negligible probability $\mathsf{CPA.Rec}$ is complete and certifying for all CPA components used below. We henceforth condition on these events.

Let $\mathcal{W}$ be the ordered list consisting of all decoding queries made by $\mathcal{A}$, followed by the final word output by $\mathcal{A}$. We first rule out double openings among words in $\mathcal{W}$. If some $\widehat{\mathsf{ct}} \in \mathcal{W}$ has a double opening at an index i, then an extractor that knows the CPA secret keys obtains $(i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho_0, \rho_1)$ such that

$$\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho_0, 0) = 1 \quad \text{and} \quad \mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho_1, 1) = 1.$$

Such normal-mode double openings are negligible by the statistical binding property of TBEQC in the normal mode.

$\mathsf{Decode}^{\mathbf{1}}$ *is correct on close transcript codewords.* Let $\mathcal{A}$ output $(j^*, \widehat{\mathsf{ct}})$. Suppose that $j^* \in [q_E]$ and $|\widehat{\mathsf{ct}} \oplus \mathsf{ct}^{(j^*)}| \le \delta L$. We first prove that $\mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}) = m^{(j^*)}$ except with negligible probability.

Call an index $i \in [N]$ good, with respect to the final transcript entry j^*, if all four local distances are within threshold:

1. $|\widehat{t}_i \oplus t_i^{(j^*)}| \le \delta_{tag} n_{tag}$,
2. $|\widehat{\mathsf{com}}_i \oplus \mathsf{com}_i^{(j^*)}| \le \delta_{com} n_{com}$,
3. $|\widehat{c}_{1,i} \oplus c_{1,i}^{(j^*)}| \le \delta_{cpa} n_{cpa}$,
4. $|\widehat{c}_{s_i^{(j^*)},i} \oplus c^{(j^*)}_{s_i^{(j^*)},i}| \le \delta_{cpa} n_{cpa}$.

Let $\Delta := \min(\delta_{tag} n_{tag}, \delta_{com} n_{com}, \delta_{cpa} n_{cpa})$. Every non-good index consumes more than Δ errors in the global Hamming budget. Therefore the number of non-good indices is at most $\frac{\delta L}{\Delta} \le \alpha N$ by the second parameter condition.

Fix a good index i.

- If $s_i^{(j^*)} = 1$, then branch 1 is the active branch and

$$c_{1,i}^{(j^*)} = \mathsf{CPA.Encode}\left(pk_{1,i}, 1 \| \rho_i^{(j^*)}; \mathsf{HPRG.Eval}(pp, s^{(j^*)}, i)\right).$$

  If $\mathsf{CPA.Decode}(sk_{1,i}, \widehat{c}_{1,i}) \neq 1 \| \rho_i^{(j^*)}$, then $\left(1 \| \rho_i^{(j^*)}, \mathsf{HPRG.Eval}(pp, s^{(j^*)}, i), \widehat{c}_{1,i}\right)$ is a witness against adaptive δ_{cpa}-robustness of the underlying CPA-secure PRC.
- If $s_i^{(j^*)} = 0$, then branch 1 is the inactive branch and

$$c_{1,i}^{(j^*)} = \mathsf{CPA.Encode}\left(pk_{1,i}, 0^{\ell_{msg}}; r_i^{(j^*)}\right).$$

  If $\mathsf{CPA.Decode}(sk_{1,i}, \widehat{c}_{1,i}) \neq 0^{\ell_{msg}}$, then $\left(0^{\ell_{msg}}, r_i^{(j^*)}, \widehat{c}_{1,i}\right)$ is a witness against adaptive δ_{cpa}-robustness.

The two decryption-error events above are negligible despite the decoder access given to $\mathcal{A}$. A reduction guesses j^* and i, embeds the CPA robustness challenge public key as $pk_{1,i}$, simulates the entire top-level experiment using the real oracle $\mathsf{Decode}^{\mathbf{0}}$, and, when $\mathcal{A}$ outputs $(j^*, \widehat{\mathsf{ct}})$, outputs $\left(Y_{1,i}^{(j^*)}, R_{1,i}^{(j^*)}, \widehat{c}_{1,i}\right)$. The simulation is exact because $\mathsf{Decode}^{\mathbf{0}}$ never uses $sk_{1,i}$. Thus, except with negligible probability, every good index is decoded correctly by $\mathsf{Find}^{\mathbf{1}}$, meaning that its output d_i equals the true bit $s_i^{(j^*)}$.

Moreover, every good index passes the final public check for the true seed $s^{(j^*)}$. Indeed, the active branch $s_i^{(j^*)}$ is within distance $\delta_{cpa} n_{cpa}$ of the honest active CPA encoding of $1\|\rho_i^{(j^*)}$ under randomness $\mathsf{HPRG.Eval}(pp, s^{(j^*)}, i)$. By completeness of certifying recovery, $\mathsf{CPA.Rec}$ returns $1\|\rho_i^{(j^*)}$, and by TBEQC robustness the corrupted tag and commitment still verify using the same opening. Hence $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s^{(j^*)}, i) = \textsc{True}$ for every good index i.

It follows that $d := \mathsf{Find}^{\mathbf{1}}(PK, SK, \widehat{\mathsf{ct}})$ differs from $s^{(j^*)}$ in at most αN positions. Thus $\mathsf{ECC.Dec}(d) = \sigma^{(j^*)}$, and the analysis decoder reconstructs $s^{(j^*)} = \mathsf{ECC.Enc}(\sigma^{(j^*)})$. Its accepting set contains every good index and therefore has size at least $(1-\alpha)N$. Finally, the core block has at most $\delta L \le \delta_{core} n_{core}$ errors, so

$$\mathsf{ECC}'.\mathsf{Dec}\left(\widehat{\mathsf{ct}}_0 \oplus \mathsf{HPRG.Eval}(pp, s^{(j^*)}, 0)\right) = m^{(j^*)}.$$

Therefore $\mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}) = m^{(j^*)}$ except with negligible probability.

$\mathsf{Decode}^{\mathbf{0}}$ *and* $\mathsf{Decode}^{\mathbf{1}}$ *agrees with overwhelming probability.* First, observe that, except with negligible probability, the following implication holds for every word $\widehat{\mathsf{ct}} \in \mathcal{W}$:

$$\mathsf{Decode}^{\mathbf{0}}(SK, \widehat{\mathsf{ct}}) = m \neq \perp \quad \Longrightarrow \quad \mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}) = m.$$

To see this, suppose $\mathsf{Decode}^{\mathbf{0}}$ accepts $\widehat{\mathsf{ct}}$ with output m. Let

$$d := \mathsf{Find}^{\mathbf{0}}(PK, SK, \widehat{\mathsf{ct}}), \qquad s := \mathsf{ECC.Enc}(\mathsf{ECC.Dec}(d)),$$

and let

$$S := \left\{ i \in [N] : d_i = s_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True} \right\}.$$

Then $|S| \ge (1-\alpha)N$. For every $i \in S$, the finder $\mathsf{Find}^{\mathbf{1}}$ also outputs s_i. If $s_i = 0$, then $d_i = 0$, so branch 0 is valid at $(\widehat{\mathsf{ct}}, i)$; since double openings have been ruled out, branch 1 is not valid, and $\mathsf{Find}^{\mathbf{1}}$ rejects branch 1 and outputs $0 = s_i$. If $s_i = 1$, then $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True}$ gives

$$y_i := \mathsf{OpenOK}_1\left(PK, \widehat{\mathsf{ct}}, i, \mathsf{HPRG.Eval}(pp, s, i)\right) \neq \perp.$$

By the branch-1 certified-error extraction above, $\mathsf{CPA.Decode}(sk_{1,i}, \widehat{c}_{1,i}) = y_i$ except with negligible probability. Therefore branch 1 is valid and $\mathsf{Find}^{\mathbf{1}}$ outputs $1 = s_i$. Hence $\mathsf{Find}^{\mathbf{1}}$ agrees with s on all indices in S, so $\mathsf{Decode}^{\mathbf{1}}$ recovers the same seed, the same core message, and accepts using the same set S. Its output is m.

For $\mathcal{A}$ to win, its final output $(j^*, \widehat{\mathsf{ct}})$ must satisfy $j^* \in [q_E]$ and

$$|\widehat{\mathsf{ct}} \oplus \mathsf{ct}^{(j^*)}| \leq \delta L, \qquad \mathsf{Decode}^{\mathbf{0}}(SK, \widehat{\mathsf{ct}}) \neq m^{(j^*)}.$$

The previous part showed that $\mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}) = m^{(j^*)}$ except with negligible probability. By the implication just proved, the $\mathsf{Decode}^{\mathbf{0}}$ cannot output a non-$\perp$ message different from $m^{(j^*)}$. Thus, except with negligible probability, a successful attack must have

$$\mathsf{Decode}^{\mathbf{0}}(SK, \widehat{\mathsf{ct}}) = \perp \quad \text{and} \quad \mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}) = m^{(j^*)}.$$

Let Diff denote the event that

$$\exists \widehat{\mathsf{ct}} \in \mathcal{W} \quad \mathsf{Decode}^{\mathbf{0}}(SK, \widehat{\mathsf{ct}}) \neq \mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}).$$

Let $\widehat{\mathsf{ct}}^{\dagger}$ denote the first word in the ordered list $\mathcal{W}$ on which the two decoders differ. By the implication above, this first disagreement must satisfy

$$\mathsf{Decode}^{\mathbf{0}}(SK, \widehat{\mathsf{ct}}^{\dagger}) = \perp \quad \text{and} \quad \mathsf{Decode}^{\mathbf{1}}(SK, \widehat{\mathsf{ct}}^{\dagger}) = m^{\dagger} \neq \perp$$

except with negligible probability.
Let $d' := \mathsf{Find}^{\mathbf{1}}(PK, SK, \widehat{\mathsf{ct}}^{\dagger}), s' := \mathsf{ECC.Enc}(\mathsf{ECC.Dec}(d'))$, and let

$$S' := \left\{ i \in [N] : d'_i = s'_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}^{\dagger}, s', i) = \textsc{True} \right\}.$$

Since $\mathsf{Decode}^{\mathbf{1}}$ accepts, $|S'| \geq (1-\alpha)N$. For $i \in S'$ with $s'_i = 1$, branch 1 is valid. Since double openings have been ruled out, branch 0 is not valid, and so $\mathsf{Find}^{\mathbf{0}}$ rejects branch 0 and outputs $1 = s'_i$. For $i \in S'$ with $s'_i = 0$, the check gives

$$y_i := \mathsf{OpenOK}_0 \left(PK, \widehat{\mathsf{ct}}^{\dagger}, i, \mathsf{HPRG.Eval}(pp, s', i) \right) \neq \perp.$$

If $\mathsf{CPA.Decode}(sk_{0,i}, \widehat{c}^{\dagger}_{0,i}) = y_i$, then branch 0 is valid and $\mathsf{Find}^{\mathbf{0}}$ outputs $0 = s'_i$. Consequently, if for all $i \in S'$ with $s'_i = 0$ one has $\mathsf{CPA.Decode}(sk_{0,i}, \widehat{c}^{\dagger}_{0,i}) \neq y_i$, then $\mathsf{Find}^{\mathbf{0}}$ agrees with s' on every index in S', and $\mathsf{Decode}^{\mathbf{0}}$ would recover the same seed, the same core message, and accept using the same set S'. This contradicts the definition of $\widehat{\mathsf{ct}}^{\dagger}$. Hence, there must exist an index $i \in S'$ with $s'_i = 0$ such that $\mathsf{OpenOK}_0 \left(PK, \widehat{\mathsf{ct}}^{\dagger}, i, \mathsf{HPRG.Eval}(pp, s', i) \right) = y_i \neq \perp$ but $\mathsf{CPA.Decode}(sk_{0,i}, \widehat{c}^{\dagger}_{0,i}) \neq y_i$. By the certifying property of OpenOK, the tuple $\left(y_i, \mathsf{HPRG.Eval}(pp, s', i), \widehat{c}^{\dagger}_{0,i} \right)$ is a witness against adaptive robustness of the underlying CPA-secure PRC.

The last issue is that the adversary has access to $\mathsf{Decode}^{\mathbf{0}}$, which uses $(sk_{i,0})_{i \in [N]}$. The reduction guesses the position of $\widehat{\mathsf{ct}}^{\dagger}$ in $\mathcal{W}$ and the index i, embeds the CPA robustness challenge public key as $pk_{0,i}$, and generates all other keys honestly. For every decoding query before the guessed first disagreement, it answers using

$\mathsf{Decode}^{\mathbf{1}}$, which does not use $sk_{0,i}$. On the event that the guess is correct, these answers are identical to $\mathsf{Decode}^{\mathbf{0}}$ answers by the minimality of $\widehat{\mathsf{ct}}^{\dagger}$. When the guessed word is reached, the reduction computes $\mathsf{Decode}^{\mathbf{1}}$, obtains s' and S', and outputs the certified tuple above for the guessed index.

Therefore $\mathbf{Pr}\,[\mathsf{Diff}]$ is negligible. Combining this with the correctness of $\mathsf{Decode}^{\mathbf{1}}$ on close transcript codewords, we conclude that $\mathcal{A}$ wins with negligible probability.

For the concrete parameters from Section 6.2, $\delta = 0.0005$ satisfies the two conditions. Indeed,

$$\delta L = 0.0005 \cdot 24 N n_{cpa} = 0.012 N n_{cpa} \le 25\delta \cdot N n_{cpa} = \delta_{core} n_{core},$$

and

$$\Delta = \min(0.03 \cdot 3n_{cpa}, 0.005 \cdot 18 n_{cpa}, 0.1 n_{cpa}) = 0.09 n_{cpa}.$$

Therefore

$$\alpha N \Delta = 0.2N \cdot 0.09 n_{cpa} = 0.018 N n_{cpa} \ge 0.012 N n_{cpa} = \delta L.$$

Hence the construction is search adaptively 0.0005-robust with decoder access.

7.3 Security Proof

We prove CCA1 pseudorandomness via a sequence of hybrids. Let $\mathcal{A}$ be a PPT CCA1 adversary making at most $q = q(\lambda)$ pre-challenge decoding queries. The challenger samples the challenge seed $\sigma^* \leftarrow \{0,1\}^{\kappa}$, sets

$$s^* := \mathsf{ECC.Enc}(\sigma^*) \in \{0,1\}^N,$$

and samples challenge tags $t_1^*, \ldots, t_N^* \leftarrow \{0,1\}^{n_{tag}}$. These values are sampled at the onset of the hybrids, before the CCA1 decoding queries are answered. This is without loss because the adversary receives the challenge ciphertext only after all decoding queries have been made.

Hybrid 0. This is the real CCA1 pseudorandomness experiment for Π.

Hybrid 1: abort on close tags. For every pre-challenge decoding query $\widehat{\mathsf{ct}} = \widehat{\mathsf{ct}}_0 \| \widehat{\mathsf{ct}}_1 \| \cdots \| \widehat{\mathsf{ct}}_N$ with $\widehat{\mathsf{ct}}_i = \widehat{t}_i \| \widehat{\mathsf{com}}_i \| \widehat{c}_{0,i} \| \widehat{c}_{1,i}$, the challenger aborts if for some $i \in [N]$, $|\widehat{t}_i \oplus t_i^*|/n_{tag} \notin [\gamma, 1-\gamma]$. We claim that the abort probability is negligible. Indeed, for fixed query j and index i, the tag $\widehat{t}_i^{(j)}$ is independent of the uniformly random t_i^*. Thus $|\widehat{t}_i \oplus t_i^*|/n_{tag} \notin [\gamma, 1-\gamma]$ implies either $|\widehat{t}_i^{(j)} \oplus t_i^*| < \gamma n_{tag}$ or $|\widehat{t}_i^{(j)} \oplus \overline{t_i^*}| < \gamma n_{tag}$. For a uniformly random t_i^*,

$$\mathbf{Pr}\left[|\widehat{t}_i^{(j)} \oplus t_i^*| \le \gamma n_{tag}\right] \le 2^{(\mathbf{H}(\gamma)-1)n_{tag}},$$

and the same bound holds for the Hamming ball around $\overline{t_i^*}$. A union bound over all q decoding queries and all N indices gives abort probability at most

$$2qN \cdot 2^{(\mathbf{H}(\gamma)-1)n_{tag}},$$

which is negligible for every constant $\gamma \in (0, 1/2)$ and $n_{tag} = \omega(\log \lambda)$.

Hybrid 2: equivocal TBEQC setup. For every $i \in [N]$, instead of generating $ck_i \leftarrow \mathsf{CKG}(1^\lambda)$ and then honestly committing to s_i^* under tag t_i^*, the challenger runs

$$(\widetilde{ck}_i, \widetilde{\mathsf{com}}_i, \widetilde{\rho}_{i,0}, \widetilde{\rho}_{i,1}) \leftarrow \mathsf{EQSetup}(1^\lambda, t_i^*).$$

It sets $ck_i := \widetilde{ck}_i$ and uses the equivocal commitment $\widetilde{\mathsf{com}}_i$ in the challenge ciphertext. The branch corresponding to s_i^* is opened using $\widetilde{\rho}_{i,s_i^*}$. Indistinguishability from Hybrid 1 follows from the mode indistinguishability of the robust TBEQC scheme, applied independently over all $i \in [N]$.

Shared notation. We use the branch-selecting decoders Decode^z and the public predicate OpenOK from Section 7.1. In particular, the original decoding oracle is $\mathsf{Decode}^{\mathbf{0}}$, and OpenOK provides the certified CPA-robustness witnesses used in the oracle-switch arguments below.

Hybrid 3: switch the finding branch to the challenge branch. Define Find^{s^*}, which takes s^* as auxiliary input. For every $i \in [N]$, it decrypts only the branch s_i^*:

$$y_i := \mathsf{CPA.Decode}(sk_{s_i^*,i}, \widehat{c}_{s_i^*,i}).$$

If $y_i = 1 \| \rho$ and $\mathsf{Verify}(ck_i, \widehat{t}_i, \widehat{\mathsf{com}}_i, \rho, s_i^*) = 1$, then Find^{s^*} sets $d_i := s_i^*$; otherwise it sets $d_i := 1 - s_i^*$. Hybrid 3 is identical to Hybrid 2 except that the decoding oracle uses Decode^{s^*} instead of $\mathsf{Decode}^{\mathbf{0}}$.

Lemma 10. *Hybrids 2 and 3 are indistinguishable.*

Proof. We prove that the two decoding oracles give identical answers except with negligible probability. The proof has two kinds of bad events. The first is a double-opening event for the robust TBEQC. The second is a CPA decryption error event; this is where adaptive robustness of the underlying CPA-secure PRC is used.

Double-opening event. For a decoding query $\widehat{\mathsf{ct}}$ and index i, let $\mathsf{Dbl}_i(\widehat{\mathsf{ct}})$ be the event that a double opening occurs at $(\widetilde{\mathsf{ct}}, i)$. Conditioned on Hybrid 1 not aborting, the tag $\widehat{t}_i$ satisfies $\gamma n_{tag} \le |\widehat{t}_i \oplus t_i^*| \le (1-\gamma) n_{tag}$. Therefore, if $\mathsf{Dbl}_i(\widehat{\mathsf{ct}})$ occurs, then $(\widehat{t}_i, \widehat{\mathsf{com}}_i, \rho_0, \rho_1)$ breaks γ-robust statistical binding in EQ mode for the TBEQC at index i. By guessing $i \in [N]$ and using the TBEQC binding challenger to generate the equivocal key for that index, any non-negligible probability of Dbl_i would contradict the robust statistical binding property. Thus, over all polynomially many decoding queries and all $i \in [N]$,

$$\mathbf{Pr}[\exists \widehat{\mathsf{ct}}, i : \mathsf{Dbl}_i(\widehat{\mathsf{ct}})] \le \mathtt{negl}(\lambda).$$

CPA decryption errors certified by Check. We next record the precise CPA robustness fact needed in the oracle-switch argument.

First consider Hybrid 2. Let DErr_2 be the event that for some decoding query $\widehat{\mathsf{ct}}$ that is accepted by the Hybrid 2 oracle, with decoded seed s and accepting set

$$S := \left\{ i \in [N] : d_i = s_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True} \right\},$$

there exists an index $i \in S$ with $s_i = 1$ such that

$$y_i \neq \perp \wedge \mathsf{CPA.Decode}(sk_{1,i}, \widehat{c}_{1,i}) \neq y_i,$$

where

$$r_i := \mathsf{HPRG.Eval}(pp, s, i) \text{ and } y_i := \mathsf{OpenOK}_1(PK, \widehat{\mathsf{ct}}, i, r_i)$$

We claim that $\mathbf{Pr}[\mathsf{DErr}_2] \leq \mathtt{negl}(\lambda)$.

Indeed, suppose $\mathbf{Pr}[\mathsf{DErr}_2]$ is non-negligible. We build an adversary $\mathcal{B}$ against adaptive δ_{cpa}-robustness of $\mathcal{E}$. The adversary $\mathcal{B}$ receives a CPA public key $pk^\dagger$ from the robustness challenger. It guesses a query number $j \in [q]$ and an index $i^\dagger \in [N]$. It embeds $pk^\dagger$ as $pk_{1,i^\dagger}$ and generates all other CPA key pairs honestly. This lets $\mathcal{B}$ answer Hybrid 2 decoding queries exactly: Hybrid 2's Find routine only uses the keys $sk_{0,i}$, and Check uses only the public recovery algorithm $\mathsf{CPA.Rec}$.

When the guessed j-th decoding query is processed, if the Hybrid 2 oracle accepts with seed s and $i^\dagger \in S$ with $s_{i^\dagger} = 1$, then $\mathcal{B}$ computes

$$r^\dagger := \mathsf{HPRG.Eval}(pp, s, i^\dagger) \quad \text{and} \quad y^\dagger := \mathsf{OpenOK}_1(PK, \widehat{\mathsf{ct}}, i^\dagger, r^\dagger).$$

If $y^\dagger \neq \perp$, then $\mathcal{B}$ outputs $(y^\dagger, r^\dagger, \widehat{c}_{1,i^\dagger})$ to the adaptive robustness challenger; otherwise it outputs a dummy tuple. Whenever DErr_2 occurs at the guessed query and index, the certifying property of $\mathsf{CPA.Rec}$ gives

$$|\widehat{c}_{1,i^\dagger} \oplus \mathsf{CPA.Encode}(pk^\dagger, y^\dagger; r^\dagger)| \leq \delta_{cpa} n_{cpa},$$

while the definition of DErr_2 gives

$$\mathsf{CPA.Decode}(sk^\dagger, \widehat{c}_{1,i^\dagger}) \neq y^\dagger.$$

Thus $\mathcal{B}$ wins the adaptive robustness game with probability at least $\mathbf{Pr}[\mathsf{DErr}_2]/(qN)$, a contradiction.

Next consider Hybrid 3. Let DErr_3 be the event that for some decoding query $\widehat{\mathsf{ct}}$ accepted by the Hybrid 3 oracle, with decoded seed s' and accepting set

$$S' := \left\{ i \in [N] : d'_i = s'_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}, s', i) = \textsc{True} \right\},$$

there exists $i \in S'$ such that

$$s'_i = 1 - s^*_i \ \wedge \ y'_i \neq \perp \ \wedge \ \mathsf{CPA.Decode}(sk_{1-s^*_i,i}, \widehat{c}_{1-s^*_i,i}) \neq y'_i,$$

where

$$r'_i := \mathsf{HPRG.Eval}(pp, s', i) \quad \text{and} \quad y'_i := \mathsf{OpenOK}_{1-s^*_i}(PK, \widehat{\mathsf{ct}}, i, r'_i).$$

Again $\mathbf{Pr}[\mathsf{DErr}_3] \leq \mathtt{negl}(\lambda)$. The reduction is the same as above, except that it embeds the robustness challenge public key at position $(1 - s^*_i, i)$ for a guessed index i. This is simulatable because Hybrid 3's Find^{s^*} routine uses only $sk_{s^*_i,i}$ and never uses $sk_{1-s^*_i,i}$; Check remains entirely public. Therefore any occurrence

of DErr_3 yields, after guessing the query and index, a CPA codeword within distance $\delta_{cpa} n_{cpa}$ of a public encoding but decrypting to a different message, contradicting adaptive robustness of $\mathcal{E}$.

Oracle equivalence conditioned on no bad event. We now condition on the complement of the three negligible events

$$\exists \widehat{\mathsf{ct}}, i : \mathsf{Dbl}_i(\widehat{\mathsf{ct}}), \qquad \mathsf{DErr}_2, \qquad \mathsf{DErr}_3.$$

Fix an arbitrary pre-challenge decoding query $\widehat{\mathsf{ct}}$. Let $Out_2(\widehat{\mathsf{ct}})$ and $Out_3(\widehat{\mathsf{ct}})$ denote the outputs of the Hybrid 2 and Hybrid 3 decoding oracles, respectively.

Suppose first that $Out_2(\widehat{\mathsf{ct}}) = m \neq \bot$. Let d be the string computed by Find in Hybrid 2, let

$$\sigma := \mathsf{ECC.Dec}(d), \qquad s := \mathsf{ECC.Enc}(\sigma),$$

and let

$$S := \left\{ i \in [N] : d_i = s_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True} \right\}.$$

Since Hybrid 2 outputs m, we have $|S| \geq (1-\alpha)N$.

We show that the string d' computed by Find^{s^*} in Hybrid 3 agrees with s on every index $i \in S$.

- If $s^*_i = 0$, then Find and Find^{s^*} both inspect branch 0 at index i. Hence they perform the same secret-key decoding and verification test, and $d'_i = d_i = s_i$.
- If $s^*_i = 1$ and $s_i = 0$, then $d_i = 0$ means that branch 0 is secretly valid: $\mathsf{CPA.Decode}(sk_{0,i}, \widehat{c}_{0,i}) = 1 \| \rho_0$ for some ρ_0, and the TBEQC verification for bit 0 accepts. If branch 1 were also secretly valid, then $\mathsf{Dbl}_i(\widehat{\mathsf{ct}})$ would occur. Since we are conditioning on no double opening, branch 1 is not secretly valid. Thus Find^{s^*}, which inspects branch 1, rejects that branch and sets $d'_i = 0 = s_i$.
- If $s^*_i = 1$ and $s_i = 1$, then $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s, i) = \textsc{True}$ certifies that branch 1 is within distance $\delta_{cpa} n_{cpa}$ of the public encoding of the opening recovered by $\mathsf{CPA.Rec}$. Since DErr_2 does not occur, this branch induces no CPA decryption error: $\mathsf{CPA.Decode}(sk_{1,i}, \widehat{c}_{1,i})$ returns the same opening. Therefore Find^{s^*} accepts branch 1 and sets $d'_i = 1 = s_i$.

Thus d' agrees with s on all indices in S, and hence on at least $(1-\alpha)N$ positions. Since ECC corrects αN errors, Hybrid 3 decodes the same seed s. The core message recovery from $\widehat{\mathsf{ct}}_0$ is therefore identical, and the same set S passes the final checking step in Hybrid 3. Hence $Out_3(\widehat{\mathsf{ct}}) = m$.

Conversely, suppose $Out_3(\widehat{\mathsf{ct}}) = m' \neq \bot$. Let d' be the string computed by Find^{s^*}, let

$$\sigma' := \mathsf{ECC.Dec}(d'), \qquad s' := \mathsf{ECC.Enc}(\sigma'),$$

and let

$$S' := \left\{ i \in [N] : d'_i = s'_i \ \wedge \ \mathsf{Check}(PK, \widehat{\mathsf{ct}}, s', i) = \textsc{True} \right\}.$$

Then $|S'| \geq (1-\alpha)N$. We show that the string d computed by Find in Hybrid 2 agrees with s' on every $i \in S'$.

- If $s_i^* = 0$, then both Find and Find^{s^*} inspect branch 0, so $d_i = d'_i = s'_i$.
- If $s_i^* = 1$ and $s'_i = 0$, then $\mathsf{Check}(PK, \widehat{\mathsf{ct}}, s', i) = \textsc{True}$ certifies that branch 0 is within distance $\delta_{cpa} n_{cpa}$ of the public encoding of the recovered opening. Since DErr_3 does not occur, branch 0 induces no CPA decryption error. Hence Find accepts branch 0 and sets $d_i = 0 = s'_i$.
- If $s_i^* = 1$ and $s'_i = 1$, then $d'_i = 1$ means that branch 1 is secretly valid. If branch 0 were also secretly valid, then $\mathsf{Dbl}_i(\widehat{\mathsf{ct}})$ would occur. Therefore branch 0 is not secretly valid, and Find rejects branch 0 and sets $d_i = 1 = s'_i$.

Thus d agrees with s' on at least $(1-\alpha)N$ positions, so Hybrid 2 decodes the same seed s', recovers the same core message, and accepts using the same valid set S'. Hence $Out_2(\widehat{\mathsf{ct}}) = m'$.

We have shown that if either oracle outputs a non-$\perp$ message, then the other oracle outputs the same message. Therefore the two decoding oracles are identical on all pre-challenge queries except with negligible probability. This proves the lemma.

Hybrid 4: simulate the un-signaled branch. For every $i \in [N]$, the branch $1 - s_i^*$ is not used by Find^{s^*}. In the challenge ciphertext, replace the encryption of $0^{\ell_{msg}}$ on branch $1-s_i^*$ by an encryption of the alternative equivocal opening $1\|\widetilde{\rho}_{i,1-s_i^*}$, using the same independently uniform randomness that was used for that branch. Since the decoding oracle in Hybrid 3 never uses the secret key $sk_{1-s_i^*,i}$, this change is indistinguishable by CPA pseudorandomness of the underlying CPA-secure PRC, applied over all $i \in [N]$.

Hybrid 5: switch to $\mathsf{Find}^{\mathbf{1}}$. The decoding oracle is changed from Decode^{s^*} to $\mathsf{Decode}^{\mathbf{1}}$, the decoder that always inspects branch 1. The same oracle-switch proof as in Lemma 10, with the roles of the two fixed branches reversed, shows that Hybrid 5 is indistinguishable from Hybrid 4. The proof again uses the robust statistical binding of the TBEQC to rule out double openings and the adaptive robustness of the CPA-secure PRC to rule out certified CPA decryption errors.

Hybrid 6: replace the HPRG outputs by uniform strings. The challenger replaces

$$K = \mathsf{HPRG.Eval}(pp, s^*, 0)$$

and the signaled branch randomness values

$$\widetilde{r}_i = \mathsf{HPRG.Eval}(pp, s^*, i) \qquad \text{for } i \in [N]$$

by uniformly random strings of the same lengths. The randomness used for the un-signaled branches is already uniform. Since $s^* = \mathsf{ECC.Enc}(\sigma^*)$ has min-entropy at least κ, the (κ, N)-robustness of the HPRG implies that Hybrid 6 is indistinguishable from Hybrid 5.

Hybrid 7: replace CPA codewords in the 0-branch by uniform strings. Now all randomness used in the CPA encodings appearing in the challenge ciphertext is uniform and independent of the adversary's pre-challenge view. Replace challenge CPA codewords $\{c^*_{0,i}\}_{i\in[N]}$ by uniformly random strings in $\{0,1\}^{n_{cpa}}$. Since $\{sk_{0,i}\}$ are not used by $\mathsf{Decode}^{\mathbf{1}}$, by the CPA pseudorandomness of $\mathcal{E}$, Hybrid 7 is indistinguishable from Hybrid 6.

Hybrid 8: switch back to $\mathsf{Decode}^{\mathbf{0}}$. This hybrid game is identical to Hybrid 7 except that the decoding oracle is switched back to $\mathsf{Decode}^{\mathbf{0}}$. This indistinguishability proof is analogous to Lemma 10.

Hybrid 9: replace CPA codewords in the 1-branch by uniform strings. Replace challenge CPA codewords $\{c^*_{1,i}\}_{i\in[N]}$ by uniformly random strings in $\{0,1\}^{n_{cpa}}$. Since $\{sk_{1,i}\}$ are not used by $\mathsf{Decode}^{\mathbf{0}}$, by the CPA pseudorandomness of $\mathcal{E}$, Hybrid 9 is indistinguishable from Hybrid 8.

Hybrid 10: replace commitments by uniform strings. Replace the challenge commitments $\widetilde{\mathsf{com}}_1, \ldots, \widetilde{\mathsf{com}}_N$ by uniformly random strings in $\{0,1\}^{n_{com}}$. By pseudorandomness of TBEQC commitments in normal mode, together with mode indistinguishability between normal and EQ setup, Hybrid 10 is indistinguishable from Hybrid 9.

At this point the entire challenge ciphertext is uniformly random: the core component is one-time padded by a uniform string, every CPA component is uniform, and every commitment component is uniform.

Hybrid 11: return to normal setup and drop the abort rule. Since the challenge ciphertext is now uniform and independent of the TBEQC mode and of the challenge tags, we switch the TBEQC keys back to normal mode and remove the abort rule introduced in Hybrid 1. The abort rule was already shown to trigger with negligible probability, and after the previous hybrids the challenge ciphertext is exactly distributed as a uniformly random codeword. Therefore Hybrid 11 is the ideal CCA1 pseudorandomness experiment.

Combining the hybrids, the real CCA1 experiment and the ideal random-codeword experiment are indistinguishable. Hence Π is CCA1-pseudorandom.

References

AAC+25. Omar Alrabiah, Prabhanjan Ananth, Miranda Christ, Yevgeniy Dodis, and Sam Gunn. Ideal pseudorandom codes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1638–1647, 2025. 2, 7, 8, 10, 19, 23, 50, 51

ABW10. Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010. 3, 12

AK19. Benny Applebaum and Eliran Kachlon. Sampling graphs without forbidden subgraphs and unbalanced expanders with negligible error. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 171–179. IEEE Computer Society, 2019. 14

BBTV25. Kiril Bangachev, Guy Bresler, Stefan Tiegel, and Vinod Vaikuntanathan. Near-optimal time-sparsity trade-offs for solving noisy linear equations. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1910–1920, 2025. 14

BFKL93. Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, Lecture Notes in Computer Science, pages 278–291. Springer, 1993. 14

BKW03. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003. 42, 43

CG24. Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pages 325–347. Springer, 2024. 2, 7, 9, 10, 23

CGG+25. Miranda Christ, Noah Golowich, Sam Gunn, Ankur Moitra, and Daniel Wichs. Improved pseudorandom codes from permuted puzzles. *arXiv preprint arXiv:2512.08918*, 2025. 2, 7

CGZ24. Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR, 2024. 7

DJ24. Quang Dao and Aayush Jain. Lossy cryptography from code-based assumptions. In *Annual International Cryptology Conference*, pages 34–75. Springer, 2024. 3, 12, 14, 42, 43, 44

DMR25. Nico Döttling, Anne Müller, and Mahesh Sreekumar Rajasree. Separating pseudorandom codes from local oracles. In *Theory of Cryptography - 23rd International Conference, TCC 2025*, Lecture Notes in Computer Science, pages 225–257. Springer, 2025. 8

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual international cryptology conference*, pages 537–554. Springer, 1999. 8

GG24. Surendra Ghentiyala and Venkatesan Guruswami. New constructions of pseudorandom codes. *arXiv preprint arXiv:2409.07580*, 2024. 2, 7, 10

GGH19. Sanjam Garg, Romain Gay, and Mohammad Hajiabadi. New techniques for efficient trapdoor functions and applications. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*, pages 33–63. Springer, 2019. 8

GGW25. Sanjam Garg, Sam Gunn, and Mingyuan Wang. Black-box crypto is useless for pseudorandom codes. In *Theory of Cryptography - 23rd International Conference, TCC 2025*, Lecture Notes in Computer Science, pages 205–224. Springer, 2025. 8

GHMO21. Sanjam Garg, Mohammad Hajiabadi, Giulio Malavolta, and Rafail Ostrovsky. How to build a trapdoor function from an encryption scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 220–249. Springer, 2021. 4, 7, 8, 11

GJL+25. Riddhi Ghosal, Aayush Jain, Paul Lou, Amit Sahai, and Neekon Vafa. Post-quantum pke from unstructured noisy linear algebraic assumptions: Beyond lwe and alekhnovichs lpn. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 64–93. Springer, 2025. 4, 8, 18

GM24. Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. In *Advances in Neural Information Processing Systems 38: NeurIPS 2024*, 2024. 7, 10

GZS25. Samuel Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *International Conference on Learning Representations*, volume 2025, pages 6612–6637, 2025. 2, 7

HKW20. Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In *Annual International Cryptology Conference*, pages 836–866. Springer, 2020. 8

Kil06. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006. 7, 8

KW19. Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In *Annual international cryptology conference*, pages 671–700. Springer, 2019. 4, 5, 8, 11

Mat25. Takahiro Matsuda. Chosen ciphertext security via bargs. In *IACR International Conference on Public-Key Cryptography*, pages 314–343. Springer, 2025. 6

MG24. 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. 2, 7

MH14. Takahiro Matsuda and Goichiro Hanaoka. Chosen ciphertext security via point obfuscation. In *Theory of Cryptography Conference*, pages 95–120. Springer, 2014. 8

MH15. Takahiro Matsuda and Goichiro Hanaoka. Constructing and understanding chosen ciphertext security via puncturable key encapsulation mechanisms. In *Theory of Cryptography Conference*, pages 561–590. Springer, 2015. 7

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing, STOC 1990*, pages 427–437, 1990. 8

Pra62. Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5):5–9, 1962. 42, 43

PW08. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 187–196, 2008. 8

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th annual symposium on foundations of computer science, FOCS 1999*, pages 543–553. IEEE, 1999. 8

SD15. Noah Stephens-Davidowitz. Solving the shortest vector problem in 2ˆ n time via discrete gaussian sampling. In *STOC'15: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 733–742. ACM, 2015. 18

A Cryptanalysis of Dense-Planted LPN

This section gives a concrete attack analysis for the Dense-Planted LPN assumption of Section 3.1. The goal is not to prove a lower bound, but to identify the security level suggested by the most standard attacks on sparse LPN [Pra62, BKW03] and dense-sparse assumptions [DJ24].

Recall the dense-planted LPN assumption. A matrix $\mathbf{M} \in \mathbb{F}_2^{m \times N}$ is sampled from $\mathcal{D}_{\mathsf{plant}}^{m,N,d,Q}$. There are hidden sets $I \subseteq [m]$ and $J \subseteq [N]$ such that

$$|I| = Q, \qquad |J| = t = Q/3,$$

and every planted row $\mathbf{M}_{i,*}$ $(i \in I)$, has Hamming weight d and support contained in J. Furthermore, for all $I' \subseteq I$ with size $|I'| \leq \sqrt{Q}$, $\mathbf{M}_{I',J}$ has full rank $|I'|$. The public matrix is

$$\mathbf{A} = \mathbf{MT} \in \mathbb{F}_2^{m \times r}, \qquad \mathbf{T} \leftarrow \mathbb{F}_2^{N \times r},$$

and the challenge vector is either $\mathbf{y} = \mathbf{As} + \mathbf{e}$, where $\mathbf{s} \leftarrow \mathbb{F}_2^r$ and $\mathbf{e} \leftarrow \mathrm{Ber}(m, \eta)$, or a uniform vector $\mathbf{u} \leftarrow \mathbb{F}_2^m$. The typical regime from Remark 1 is

$$N = \lambda,\ m = N^{1.1},\ Q = 3\log N,\ t = \log N,\ d = (\log N)^{1/4},\ r = \alpha N$$

for a constant $\alpha \in (0, 1)$ and constant noise $\eta \in (0, 1/2)$.

A.1 Dual Attack

Since our noise rate is constant, the dual attack aims to find a sparse vector $\mathbf{v} \in \ker(\mathbf{A})$ with $|\mathbf{v}| = O(\log \lambda)$. Such a vector $\mathbf{v}$ will make $\mathbf{v}^\top \mathbf{y} = \mathbf{v}^\top \mathbf{e}$ have a bias of $1/\mathtt{poly}(\lambda)$ while $\mathbf{v}^\top \mathbf{u}$ is a uniform bit when $\mathbf{u} \leftarrow \mathbb{F}_2^m$.

Since $\ker(\mathbf{M}^\top) \subseteq \ker(\mathbf{A}^\top)$, the dual distance of $\mathbf{A}$ is at most the dual distance of $\mathbf{M}$, which is at most $Q = \log \lambda$. We claim that with overwhelming probability, the dual distance of $\mathbf{A}$ equals that of $\mathbf{M}$. Indeed, for a vector $\mathbf{z}$ with $|\mathbf{z}| < \log \lambda$, we have $|\mathbf{z}^\top \mathbf{M}| < |\mathbf{z}| \cdot d < \log^{5/4} \lambda$; however,

$$\Pr_{\mathbf{T} \leftarrow \mathbb{F}_2^{N \times r}} \left[\exists \mathbf{z}' \in \ker(\mathbf{T}^\top) \setminus \{\mathbf{0}\} \text{ s.t. } |\mathbf{z}'| < \log^{5/4} \lambda \right] = \mathtt{negl}(\lambda).$$

Therefore, with overwhelming probability, for all $\mathbf{z} \in \ker(\mathbf{T}^\top)$, either $\mathbf{z}^\top \mathbf{M} = \mathbf{0}$ or $|\mathbf{z}| > \log \lambda$.

Our choice of $\mathbf{M}$ guarantees that the dual distance of $\mathbf{M}_{I,*}$ is at least $\sqrt{Q}$, and because other rows are independently, uniformly sampled d-sparse vectors, the dual distance of $\mathbf{M}$ is at least $\sqrt{Q}$ with overwhelming probability. This rules out easy attacks such as duplicate rows or any dependency detectable by enumerating subsets of rows of size $\sqrt{Q}$.

Therefore, though there exists $\mathbf{v} \in \ker(\mathbf{A}^\top)$ with $|\mathbf{v}| = O(\log \lambda)$, but it must be that $|\mathbf{v}| > \sqrt{\log \lambda}$. However, for an arbitrary dense matrix, we do not know a better way to find such a sparse kernel vector than brute force. A brute force search requires at least $O(N^{\sqrt{Q}}) = O(\lambda^{\sqrt{\log \lambda}})$ time, which is quasipolynomial.

A.2 Generic LPN and ISD-type Algorithms

Generic attacks ignore the distribution of $\mathbf{A}$ and treat the sample as a standard LPN instance with secret dimension $r = \alpha N$, sample count $m = N^{1.1}$, and noise η. The information-set decoding attack [Pra62] guesses r noiseless equations and solves for $\mathbf{s}$. For constant η, the probability that a guessed information set is clean is

$$(1-\eta)^r = 2^{-\Theta_\eta(N)},$$

so the basic ISD cost is exponential in N. More sophisticated generic LPN algorithms, including BKW-type and statistical-decoding variants [BKW03], improve constants and tradeoffs in various low-noise or sample-rich regimes; they do not approach the quasi-polynomial attack below for the present constant-noise, polynomial-sample parameters.

A.3 Recovering the Hidden Factors M, T

A final attack family tries to recover enough of the factorization $\mathbf{A} = \mathbf{MT}$ to expose the sparse matrix $\mathbf{M}$. This is the analogue of the factor-recovery discussion for Dense-Sparse LPN [DJ24, Sec. 7.2.1].

When $r < N$, a natural workaround is to try to invert an unknown square submatrix of $\mathbf{T}$. Let $R \subseteq [N]$ have size r, and suppose $\mathbf{T}_{R,*} \in \mathbb{F}_2^{r\times r}$ is invertible. If an attacker knew R and $\mathbf{Z} = \mathbf{T}_{R,*}^{-1}$, then

$$\mathbf{AZ} = \mathbf{MTZ} = \mathbf{M}_{*,R} + \mathbf{M}_{*,\overline{R}}\mathbf{T}_{\overline{R},*}\mathbf{Z}.$$

A row of $\mathbf{M}$ whose support is contained in R remains sparse; a row that touches $\overline{R}$ receives a dense random contribution and becomes statistically close to a dense row. The survival probability of a random d-sparse row is

$$\mathbf{Pr}[\mathrm{supp}(\mathbf{M}_{i,*}) \subseteq R] = \frac{\binom{r}{d}}{\binom{N}{d}} = \alpha^d(1+o(1)) = \alpha^{(\log N)^{1/4}}(1+o(1)) = N^{-o(1)}.$$

Thus, under a correct unmasking, many random sparse rows would survive: $m\alpha^d = N^{1.1-o(1)}$. For the planted block, the stronger event $J \subseteq R$ has probability

$$\frac{\binom{N-t}{r-t}}{\binom{N}{r}} = \alpha^t(1+o(1)) = N^{-\log_2(1/\alpha)+o(1)},$$

which is only inverse-polynomial. The difficulty is not that a useful submatrix is too rare; it is that the attacker does not know R, $\mathbf{T}_{R,*}$, or $\mathbf{Z}$. Guessing R costs

$$\binom{N}{r} = 2^{H(\alpha)N+o(N)},$$

where H is the binary entropy function, and searching directly over invertible $r \times r$ matrices is larger. Equivalently, the attacker sees many dense vectors in $\mathbb{F}_2^r$ that are sparse combinations of an unknown N-element dictionary; recovering

a basis in which many rows become sparse resembles a noisy binary dictionary-learning problem with no evident polynomial-time algorithm.

Algebraic recovery appears no easier. One may introduce variables for $\mathbf{M}$ and $\mathbf{T}$, impose $\mathbf{A} = \mathbf{MT}$, Boolean constraints, row-weight constraints $|\mathbf{M}_{i,*}| = d$, and the existence of a hidden block (I, J). These constraints are high-degree and highly symmetric. This is unlike Arora–Ge style attacks on special LPN noise models, where each sample gives a low-degree algebraic condition. The known Dense-Sparse LPN analysis reaches the same conclusion: natural factor-recovery approaches are near-exponential when the dense mask is compressing [DJ24].

B Construction of Robust Tag-Based Equivocal Bit Commitment

In this section, we present the construction of a Robust Tag-Based Equivocal Bit Commitment (Robust TBEQC) scheme.

Building blocks.

- **Pseudorandom Generator (PRG):** A standard PRG $G : \{0,1\}^{\ell_{open}} \to \{0,1\}^{n_{com}}$.
- **Random Index Hashing:** A family of hash functions $\mathcal{H}$ from $\{0,1\}^{n_{tag}}$ to $\{0,1\}^{n_{com}}$. A hash function $h_B \in \mathcal{H}$ is keyed by a sequence of indices $B = (B_1, \ldots, B_{n_{com}}) \in [n_{tag}]^{n_{com}}$ and is defined as $h_B(t) \stackrel{\mathsf{def}}{=} (t_{B_1}, \ldots, t_{B_{n_{com}}})$. Note that h_B is linear over $\mathbb{F}_2$, namely, $h_B(x \oplus y) = h_B(x) \oplus h_B(y)$.

Let $\tau := \delta_{\mathsf{com}} + \delta_{tag} + \epsilon$ be the relative verification threshold where $\varepsilon > 0$ is any constant (as long as $\tau < 1$). The Robust TBEQC scheme $\mathcal{C}$ consisting of $(\mathsf{CKG}, \mathsf{Com}, \mathsf{Verify}, \mathsf{EQSetup})$ is constructed as follows:

Key Generation $\mathsf{CKG}(1^\lambda)$*:*

1. Sample a uniform masking string $A \leftarrow \{0,1\}^{n_{com}}$.
2. Sample a sequence of random indices $B \leftarrow [n_{tag}]^{n_{com}}$ that defines the hash function h_B.
3. Output the commitment key $ck = (A, B)$.

Commitment $\mathsf{Com}(ck, t, s; \rho)$*:*

1. Parse $ck = (A, B)$.
2. If $s = 0$, output $\mathsf{com} := G(\rho)$; if $s = 1$, output $\mathsf{com} := G(\rho) \oplus A \oplus h_B(t)$.

Verification $\mathsf{Verify}(ck, \widehat{t}, \widehat{\mathsf{com}}, \rho, s)$*:*

1. Parse $ck = (A, B)$.
2. Compute $\mathsf{com}' := G(\rho) \oplus s \cdot (A \oplus h_B(\widehat{t}))$.
3. If $|\widehat{\mathsf{com}} \oplus \mathsf{com}'| \le \tau \cdot n_{com}$, output 1. Otherwise, output 0.

Equivocal Setup $\mathsf{EQSetup}(1^\lambda, t^*)$*:*

1. Sample $B \leftarrow [n_{tag}]^{n_{com}}$.
2. Sample opening randomness $\widetilde{\rho}_0, \widetilde{\rho}_1 \leftarrow \{0,1\}^{\ell_{open}}$.
3. Compute the equivocating mask $A = G(\widetilde{\rho}_0) \oplus G(\widetilde{\rho}_1) \oplus h_B(t^*)$.
4. Set the equivocal key $\widetilde{ck} = (A, B)$ and the equivocal commitment $\widetilde{\mathsf{com}} = G(\widetilde{\rho}_0)$.
5. Output $(\widetilde{ck}, \widetilde{\mathsf{com}}, \widetilde{\rho}_0, \widetilde{\rho}_1)$.

In the following, we shall prove Theorem 11 by showing that the above construction is a $(\delta_{com}, \delta_{tag}, \gamma, n_{com}, n_{tag}, \ell_{open})$-robust TBEQC, provided that

- $\log \frac{1}{1-\gamma} > 2\,\mathbf{H}(\delta_{com} + \delta_{tag} + \epsilon)$, and
- $n_{com} \geq \max\left\{\frac{n_{tag}+\ell_{open}+\lambda}{\zeta}, \quad \frac{\ln 2 \cdot \mathbf{H}(\delta_{tag}) \cdot n_{tag}}{\epsilon}\right\}$ where $\zeta = \log \frac{1}{1-\gamma} - 2\,\mathbf{H}(\delta_{com} + \delta_{tag} + \epsilon)$.

$(\delta_{com}, \delta_{tag})$-Robust Verification. Let $t \in \{0,1\}^{n_{tag}}, s \in \{0,1\}, \rho \in \{0,1\}^{\ell_{open}}$ and suppose that $\mathsf{com} := \mathsf{Com}(ck, t, s; \rho)$ is generated honestly, and let $\widehat{\mathsf{com}} = \mathsf{com} \oplus e$ and $\widehat{t} = t \oplus e'$ for some adversarial errors where $|e| \leq \delta_{com} n_{com}$ and $|e'| \leq \delta_{tag} n_{tag}$.

During verification, the receiver computes $\mathsf{com}' = G(\rho) \oplus s \cdot (A \oplus h_B(\widehat{t}))$. Notice that the original commitment components satisfy $G(\rho) \oplus s \cdot A = \mathsf{com} \oplus s \cdot h_B(t)$. Thus,

$$\mathsf{com}' = \mathsf{com} \oplus s \cdot (h_B(t) \oplus h_B(t \oplus e')) = \mathsf{com} \oplus s \cdot h_B(e')$$

where the final equality leverages the $\mathbb{F}_2$-linearity of the bit-sampling LSH.

The verification passes if $|\widehat{\mathsf{com}} \oplus \mathsf{com}'| \leq \tau n_{com}$. We bound the distance as follows:

$$\begin{aligned} |\widehat{\mathsf{com}} \oplus \mathsf{com}'| &= |(\mathsf{com} \oplus e) \oplus (\mathsf{com} \oplus s \cdot h_B(e'))| \\ &= |e \oplus s \cdot h_B(e')| \\ &\leq |e| + s \cdot |h_B(e')| \\ &\leq \delta_{com} n_{com} + |h_B(e')|. \end{aligned}$$

Recall the threshold is set to $\tau = \delta_{com} + \delta_{tag} + \epsilon$. For verification to fail, it must be that $|h_B(e')| > (\delta_{tag} + \epsilon) n_{com}$.

Since h_B samples n_{com} independent coordinates from e' uniformly with replacement, the expected weight of the hashed error is exactly $p \cdot n_{com}$, where $p = |e'|/n_{tag} \leq \delta_{tag}$. Let $X = |h_B(e')|$. By Hoeffding's inequality, for any *fixed* error vector e':

$$\mathop{\mathbf{Pr}}_{B}[X > (\delta_{tag} + \epsilon) n_{com}] \leq \mathop{\mathbf{Pr}}_{B}[X > (p + \epsilon) n_{com}] \leq \exp(-2\epsilon^2 n_{com}).$$

However, the adversary is allowed to choose e' adaptively after seeing the public key ck, which contains B. We must ensure verification holds simultaneously for *all* valid tag errors e'. The number of possible tag errors of weight

at most $\delta_{tag} n_{tag}$ is bounded by $\sum_{i=0}^{\delta_{tag} n_{tag}} \binom{n_{tag}}{i} \leq 2^{\mathbf{H}(\delta_{tag}) n_{tag}}$. Taking a union bound over all such valid e':

$$\Pr_B[\exists e' : |h_B(e')| > (\delta_{tag} + \epsilon) n_{com}] \leq 2^{\mathbf{H}(\delta_{tag}) n_{tag}} \cdot e^{-2\epsilon^2 n_{com}}.$$

By ensuring the commitment length n_{com} is chosen to be sufficiently large relative to the tag length n_{tag} such that

$$n_{com} \geq \frac{\mathbf{H}(\delta_{tag}) n_{tag} \ln 2 + \lambda}{\epsilon},$$

this failure probability is bounded by $2^{-\lambda}$. Thus, the verification condition holds for all valid errors with overwhelming probability.

Pseudorandomness of Commitments In normal mode, $G(\rho)$ is computationally indistinguishable from uniform by the security of the PRG.

Mode Indistinguishability In normal mode, A is sampled uniformly. In EQ mode, $A = G(\widetilde{\rho}_0) \oplus G(\widetilde{\rho}_1) \oplus h_B(t^*)$. Because $\widetilde{\rho}_1$ is strictly hidden from the adversary's view during the setup phase, the term $G(\widetilde{\rho}_1)$ acts as a pseudorandom pad, rendering the simulated A computationally indistinguishable from a true uniform string.

γ-Robust Statistical Binding Suppose an adversary finds valid openings to both bits for a corrupted tag $\widehat{t}$. We have:

$$\begin{aligned}
&\mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_0, 0) = 1 \wedge \mathsf{Verify}(\widetilde{ck}, \widehat{t}, \widehat{\mathsf{com}}, \rho_1, 1) = 1 \\
\implies &\exists e_0, e_1 \in \{0,1\}^{n_{\mathsf{com}}} \text{ s.t. } |e_0|, |e_1| \leq \tau n_{com} \text{ and } \mathsf{com}_0 \oplus e_0 = \widehat{\mathsf{com}} = \mathsf{com}_1 \oplus e_1 \\
\implies &\exists e_0, e_1 \in \{0,1\}^{n_{\mathsf{com}}} \text{ s.t. } |e_0|, |e_1| \leq \tau n_{com} \text{ and } h_B(t^*) \oplus h_B(\widehat{t}) = hv
\end{aligned}$$

where

$$\begin{aligned}
\mathsf{com}_0 &= G(\rho_0), \\
\mathsf{com}_1 &= G(\rho_1) \oplus A \oplus h_B(\widehat{t}) \;=\; G(\rho_1) \oplus G(\widetilde{\rho}_0) \oplus G(\widetilde{\rho}_1) \oplus h_B(t^*) \oplus h_B \\
hv &= e_0 \oplus e_1 \oplus G(\rho_0) \oplus G(\rho_1) \oplus G(\widetilde{\rho}_0) \oplus G(\widetilde{\rho}_1).
\end{aligned}$$

Fix $\widetilde{\rho}_0, \widetilde{\rho}_1, t^*$ and only consider the random choice of B. For any tag $\widehat{t}$ such that $|\widehat{t} \oplus t^*|/n_{tag} \in [\gamma, 1-\gamma]$, consider the random variable

$$X \stackrel{\mathsf{def}}{=} h_B(\widehat{t}) \oplus h_B(t^*) = h_B(\widehat{t} \oplus t^*),$$

where the second equality follows from the fact that h_B is linear. The bits of X are i.i.d. samples from $\mathsf{Ber}(\eta)$ where $\eta \stackrel{\mathsf{def}}{=} |\widehat{t} \oplus t^*|/n_{tag} \in [\gamma, 1-\gamma]$. Therefore, for any fixed vector hv,

$$\Pr_B[X = hv] \leq (\max\{\eta, 1-\eta\})^{n_{com}} \leq (1-\gamma)^{n_{com}}.$$

Next, we bound the number of possible values hv can take. Since $\widetilde{\rho}_0, \widetilde{\rho}_1$ are fixed during setup, hv depends solely on the choices of ρ_0, ρ_1, e_0, e_1. Because $|e_0| \leq \tau n_{com}$the number of choices for the pair e_0 is bounded by $\sum_{i=0}^{\tau n_{com}} \binom{n_{com}}{i} \leq 2^{\mathbf{H}(\tau) n_{com}}$; same for e_1. The number of choices for the pair (ρ_0, ρ_1) is $2^{2\ell_{open}}$.

Finally, we take a union bound over all possible $\widehat{t}$ (at most $2^{n_{tag}}$ choices), all ρ_0, ρ_1, and all valid error vectors e_0, e_1. The event that there exists ***any*** such tuple breaking the binding property happens with probability at most

$$p \leq 2^{n_{tag}} \cdot 2^{2\ell_{open}} \cdot 2^{2\,\mathbf{H}(\tau) n_{com}} \cdot (1-\gamma)^{n_{com}}.$$

Therefore, if

$$(\log \frac{1}{1-\gamma}) \cdot n_{com} \geq n_{tag} + 2\ell_{open} + \lambda + 2\,\mathbf{H}(\tau) n_{com},$$

or equivalently

$$n_{com} \geq \frac{n_{tag} + 2\ell_{open} + \lambda}{\log \frac{1}{1-\gamma} - 2\,\mathbf{H}(\tau)},$$

then $p \leq 2^{-\lambda}$.

Statistical binding in the normal mode. This follows from the γ-robust statistical binding property in the equivocal mode (with a uniformly random sampled targeted tag t^*) and mode indistinguishability.

C Other Omitted Proofs

C.1 Security Analysis of the Candidate Construction

Lemma 11 (Local distribution under gadget noise). *Let* $\mathbf{c} \leftarrow \mathsf{Enc}(\mathsf{pk})$ *and let* $\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{G}_m \mathbf{f} \pmod q$*, where* $\mathbf{f} \in \{0,1\}^{m\ell_q}$ *is independent of the hidden set* I*. Then*

$$(\widetilde{\mathbf{c}} - \mathbf{z})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J + \mathbf{e}_I + \mathbf{G}_Q \mathbf{f}_I \pmod q,$$

where $\mathbf{f}_I \in \{0,1\}^{Q\ell_q}$ *is the restriction of* $\mathbf{f}$ *to the blocks indexed by* I*. If* $|\mathbf{f}| = D_{\mathsf{err}}$*, then* $|\mathbf{f}_I|$ *has the hypergeometric distribution obtained by sampling* $Q\ell_q$ *positions without replacement from a population of size* $m\ell_q$ *containing* D_{err} *marked positions.*

Proof. Since $\mathbf{c} = \mathbf{MTs} + \mathbf{e} + \mathbf{z}$, subtracting $\mathbf{z}$ and restricting to rows in I gives

$$(\widetilde{\mathbf{c}} - \mathbf{z})_I = (\mathbf{MTs})_I + \mathbf{e}_I + (\mathbf{G}_m \mathbf{f})_I.$$

For all planted rows $i \in I$, the support of $\mathbf{M}_{i,*}$ is contained in J. Hence

$$(\mathbf{MTs})_I = \mathbf{M}_{I,J}(\mathbf{Ts})_J.$$

Moreover, gadget recomposition commutes with restricting to a set of coordinates, so $(\mathbf{G}_m \mathbf{f})_I = \mathbf{G}_Q \mathbf{f}_I$. The hypergeometric claim follows because I is a uniformly random Q-subset independent of $\mathbf{f}$.

Theorem 13 (Robustness). *For every ρ-bounded channel $\mathcal{E}$,*

$$\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}(1^\lambda),\mathbf{c}\leftarrow\mathsf{Encode}(pk)}\left[\mathsf{Dec}(sk,\mathcal{E}(\mathbf{c}))=1\right]\geq 1-\mathtt{negl}(\lambda).$$

Proof. By Lemma 11, the local word has the form

$$\mathbf{y}=\mathbf{C}\mathbf{v}+\mathbf{e}_I+\mathbf{G}_Q\mathbf{f}_I \pmod q, \qquad \mathbf{C}=\mathbf{M}_{I,J}, \quad \mathbf{v}=(\mathbf{T}\mathbf{s})_J.$$

The internal error bound $\|\mathbf{e}_I\|<B_{\mathsf{in}}$ holds with probability at least $1-1/\mathtt{poly}(N)$ by the definition of B_{in}. Conditioned on the global channel event $|\mathbf{f}|\leq\rho m\ell_q$, the random variable $|\mathbf{f}_I|$ is hypergeometric with expectation at most $\rho Q\ell_q$. By Lemma 4,

$$\mathbf{Pr}\left[|\mathbf{f}_I|>(\rho+\varepsilon)Q\ell_q\right]\leq\exp(-\Omega(\varepsilon^2Q\ell_q))=\mathtt{negl}(\lambda).$$

On the intersection of these two good events, $(\mathbf{C},\mathbf{y})$ is a promised yes-instance of $\mathsf{MED}_{t,Q,q,B_{\mathsf{in}},B_{\mathsf{out}}}$. Therefore $\mathsf{SolveMED}$ outputs a valid witness, and the verifier in the decoder accepts.

For soundness, define the two local error volumes

$$V_{\mathsf{sm}}(Q,q,B)\stackrel{\mathsf{def}}{=}\#\{\mathbf{e}\in\mathbb{Z}_q^Q:\|\mathbf{e}\|_\infty<B\}, \qquad V_{\mathsf{bit}}(Q\ell_q,B')\stackrel{\mathsf{def}}{=}\sum_{i=0}^{B'}\binom{Q\ell_q}{i}.$$

Theorem 14 (Soundness). *For every fixed $\mathbf{u}\in\mathbb{Z}_q^m$, Construction 2 has weak soundness error*

$$\Pr_{(pk,sk)\leftarrow\mathsf{KeyGen}}\left[\mathsf{Dec}(sk,\mathbf{u})=1\right]\leq q^{t-Q}\cdot V_{\mathsf{sm}}(Q,q,B_{\mathsf{in}})\cdot V_{\mathsf{bit}}(Q\ell_q,B_{\mathsf{out}}).$$

In particular, with our parameter setting, the scheme has a negligible weak soundness error.

Proof. Fix $\mathbf{u}$. In the key generation, since $\mathbf{z}\leftarrow\mathbb{Z}_q^m$ is uniform and independent of $(I,J,\mathbf{M})$, the local vector $(\mathbf{u}-\mathbf{z})_I$ is uniform over $\mathbb{Z}_q^Q$, even after conditioning on $(I,J,\mathbf{M}_{I,J})$. The decoder can accept only if this uniform vector lies in $\mathsf{dMED}_{\mathbf{C},q,B_{\mathsf{in}},B_{\mathsf{out}}}$, where $\mathbf{C}=\mathbf{M}_{I,J}$. For any fixed $\mathbf{C}$, the number of possible accepted local values is at most $q^t\cdot V_{\mathsf{sm}}(Q,q,B_{\mathsf{in}})\cdot V_{\mathsf{bit}}(Q\ell_q,B_{\mathsf{out}})$, because there are q^t choices for $\mathbf{v}$, at most $V_{\mathsf{sm}}(Q,q,B_{\mathsf{in}})$ choices for the small error, and at most $V_{\mathsf{bit}}(Q\ell_q,B_{\mathsf{out}})$ choices for the binary sparse error. Dividing by q^Q proves the stated bound.

In our parameter setting, $q=\lambda^6, V_{\mathsf{sm}}(Q,q,B_{\mathsf{in}})=\lambda^Q$. Since $\rho+\varepsilon<1/10$,

$$V_{\mathsf{bit}}(Q\ell_q,B_{\mathsf{out}})\leq 2^{Q\ell_q\,\mathbf{H}(\rho+\varepsilon)}<q^{0.47Q}.$$

Consequently,

$$q^t\cdot V_{\mathsf{sm}}(Q,q,B_{\mathsf{in}})\cdot V_{\mathsf{bit}}(Q\ell_q,B_{\mathsf{out}})<\lambda^{2Q}\cdot\lambda^Q\cdot\lambda^{2.82}=\lambda^{5.82},$$

meaning that the soundness error is negligible.

Theorem 15 (Pseudorandomness). *If* $\mathsf{DPLWE}(m, N, r, d, Q, q, \chi)$ *holds, then Construction 2 is pseudorandom.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary in the PRC pseudorandomness experiment, and suppose that it makes at most $L = \mathtt{poly}(\lambda)$ encoding queries. In the real experiment, the public key is $(\mathbf{A}, \mathbf{z}) = (\mathbf{MT}, \mathbf{z})$, and the matrix of all responses is

$$\mathbf{C}_{\mathsf{real}} = \mathbf{AS} + \mathbf{E} + \mathbf{z}\mathbf{1}_L^\top = \mathbf{MTS} + \mathbf{E} + \mathbf{z}\mathbf{1}_L^\top,$$

where $\mathbf{S} \leftarrow \mathbb{Z}_q^{r \times L}, \mathbf{E} \leftarrow \chi^{m \times L}$. By the matrix form of Assumption 2,

$$(\mathbf{MT}, \mathbf{MTS} + \mathbf{E}) \approx_c (\mathbf{MT}, \mathbf{U}),$$

where $\mathbf{U} \leftarrow \mathbb{Z}_q^{m \times L}$ is uniform. Since $\mathbf{z}$ is independently sampled and public, adding $\mathbf{z}\mathbf{1}_L^T$ to the second component is efficient post-processing, so

$$(\mathbf{A}, \mathbf{z}, \mathbf{AS} + \mathbf{E} + \mathbf{z}\mathbf{1}_L^\top) \approx_c (\mathbf{A}, \mathbf{z}, \mathbf{U} + \mathbf{z}\mathbf{1}_L^\top).$$

For every fixed $(\mathbf{A}, \mathbf{z})$, the matrix $\mathbf{U} + \mathbf{z}\mathbf{1}_L^T$ is uniform over $\mathbb{Z}_q^{m \times L}$. This is exactly the random branch of the PRC experiment.

Proof of Theorem 9. Weak robustness follows from Theorem 13, weak soundness from Theorem 14, and pseudorandomness from Theorem 15.

C.2 Equivalence of Search and Decisional Robustness

Lemma 12 (Lemma 9 restated). *Search adaptive δ-robustness with decoder access and decisional adaptive δ-robustness with decoder access are equivalent up to a polynomial loss.*

Proof. For a current transcript $Q = ((m_1, c_1), \ldots, (m_t, c_t))$ and a decoding query x, say that x is bad if

$$\exists i \in Q_\delta(x) \quad \text{such that} \quad \mathsf{Decode}(sk, x) \neq m_i.$$

A bad query immediately gives a valid search witness. Conversely, if x is not bad, then the real and patched decoding oracles give the same answer on x: this is immediate when $Q_\delta(x) = \emptyset$, and otherwise $\mathsf{Decode}(sk, x) = m_i$ for every $i \in Q_\delta(x)$, so $\mathsf{Patch}_{Q,\delta}(x)$ always returns the same value.

Let $\mathcal{A}$ be a decisional adversary. Construct a search adversary $\mathcal{B}$ that runs $\mathcal{A}$ in the real search experiment, forwards all oracle queries, and records the encoding transcript Q. On a decoding query x, let $y = \mathsf{Decode}(sk, x)$ be the answer returned by the search experiment. If there is an index $i \in Q_\delta(x)$ with $y \neq m_i$, then $\mathcal{B}$ outputs (i, x); otherwise it continues the simulation.

Couple the real and patched decisional experiments using the same keys, encoding randomness, adversary coins, and oracle randomness. By the observation above, the two executions are identical until the first bad decoding query. Therefore

$$\mathbf{Adv}^{\mathsf{drob\text{-}cca}}_{\mathcal{A},\mathsf{PRC},\delta}(\lambda) \leq \mathbf{Pr}[\mathcal{A} \text{ makes a bad decoding query}] \leq \mathbf{Adv}^{\mathsf{srob\text{-}cca}}_{\mathcal{B},\mathsf{PRC},\delta}(\lambda).$$

Thus decisional robustness follows from search robustness.

For the converse, let $\mathcal{A}'$ be a search adversary. Construct a decisional adversary $\mathcal{B}'$ that runs $\mathcal{A}'$ using its own encoding and decoding oracles and records the transcript Q. When $\mathcal{A}'$ outputs $(i^*, \widehat{c})$, the adversary $\mathcal{B}'$ first checks whether $i^* \in Q_\delta(\widehat{c})$. If not, it outputs 0. Otherwise, writing (m_{i^*}, c_{i^*}) for the corresponding transcript entry, it makes

$$T = (|Q| + 1)(\lambda + 1)$$

additional decoding queries equal to $\widehat{c}$, and outputs 1 iff none of the answers is equal to m_{i^*}.

In the real-decoder experiment, $\mathcal{B}'$ perfectly simulates the search experiment for $\mathcal{A}'$. If $\mathcal{A}'$ outputs a valid search witness, then $\mathsf{Decode}(sk, \widehat{c}) \neq m_{i^*}$, so all of the final repeated queries return a value different from m_{i^*}. Hence

$$\mathbf{Pr}\big[\mathcal{G}^{\mathsf{drob\text{-}cca}}_{\mathcal{B}',\mathsf{PRC},\delta}(1^\lambda, 0) = 1\big] \geq \mathbf{Adv}^{\mathsf{srob\text{-}cca}}_{\mathcal{A}',\mathsf{PRC},\delta}(\lambda).$$

In the patched experiment, condition on any view of $\mathcal{A}'$ and any output $(i^*, \widehat{c})$ passing the check $i^* \in Q_\delta(\widehat{c})$. Each final patched query samples an index from $Q_\delta(\widehat{c})$, and therefore returns m_{i^*} with probability at least $1/|Q|$. Thus

$$\mathbf{Pr}[\mathcal{B}' \text{ outputs } 1] \leq \left(1 - \frac{1}{|Q|}\right)^T \leq e^{-(\lambda+1)}.$$

Consequently,

$$\mathbf{Adv}^{\mathsf{drob\text{-}cca}}_{\mathcal{B}',\mathsf{PRC},\delta}(\lambda) \geq \mathbf{Adv}^{\mathsf{srob\text{-}cca}}_{\mathcal{A}',\mathsf{PRC},\delta}(\lambda) - e^{-(\lambda+1)}.$$

The reduction uses only polynomially many additional decoding queries. Hence the two notions are equivalent up to polynomial loss.

C.3 Recovery from Randomness

We recall the public-key rate-boosting transformation from [AAC+25, Section 6.2] and observe that it satisfies the recovery from the randomness property.

The transformation. Let $\Pi_1 = (\Pi_1.\mathsf{KeyGen}, \Pi_1.\mathsf{Encode}, \Pi_1.\mathsf{Decode})$ be a single-bit public-key PRC with codewords of length n_1 that is adaptively robust up to error rate $\beta \in (0, 1/4)$. Let

$$\mathsf{ECC}_1 = (\mathsf{Enc}_1, \mathsf{Dec}_1), \qquad \mathsf{ECC}_2 = (\mathsf{Enc}_2, \mathsf{Dec}_2)$$

be deterministic error-correcting codes, where

$$\mathsf{Enc}_1 : \{0,1\}^\lambda \to \{0,1\}^\ell, \qquad \mathsf{Enc}_2 : \{0,1\}^k \to \{0,1\}^{\ell n_1},$$

and suppose that ECC_2 corrects up to $\alpha \ell n_1$ worst-case errors. Let $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{\ell n_1}$ be a standard PRG. The boosted public-key PRC Π uses the same

public/secret keys as Π_1. To encode $m \in \{0,1\}^k$, the encoder samples a seed $s \leftarrow \{0,1\}^\lambda$, computes

$$s' = \mathsf{Enc}_1(s) = (s'_1, \ldots, s'_\ell) \in \{0,1\}^\ell,$$

samples independent coins $\rho_1, \ldots, \rho_\ell$ for the single-bit encoder, and forms

$$x_i := \Pi_1.\mathsf{Encode}(pk, s'_i; \rho_i) \in \{0,1\}^{n_1} \qquad (i \in [\ell]),$$

together with the masked payload

$$y := \mathsf{Enc}_2(m) \oplus \mathsf{PRG}(s) \in \{0,1\}^{\ell n_1}.$$

The final codeword is $c := x_1 \| \cdots \| x_\ell \| y$. which is of length $N = 2\ell n_1$.

Recovery algorithm. We now define a recovery algorithm that recovers the message using the encoder randomness. The full encoder randomness is

$$R = (s, \rho_1, \ldots, \rho_\ell).$$

Given $(pk, \widehat{c}, R)$, the recovery algorithm parses $\widehat{c} = \widehat{x}_1 \| \cdots \| \widehat{x}_\ell \| \widehat{y}$. Since the seed s is already given as part of R, the recovery algorithm does not need to decode the first ℓ blocks. It simply unmasks the payload and decodes:

$$m^* := \mathsf{Dec}_2\big(\widehat{y} \oplus \mathsf{PRG}(s)\big).$$

If $m^* = \perp$, it returns $\perp$. It then re-encodes with the same randomness

$$\begin{aligned} c^* &:= \Pi.\mathsf{Encode}(pk, m^*; R) \\ &= \Pi_1.\mathsf{Encode}(pk, s'_1; \rho_1) \| \cdots \| \Pi_1.\mathsf{Encode}(pk, s'_\ell; \rho_\ell) \| \big(\mathsf{Enc}_2(m^*) \oplus \mathsf{PRG}(s)\big), \end{aligned}$$

and returns m^* only if $|\widehat{c} \oplus c^*| \le \delta_{rec} N$. for some threshold δ_{rec}; otherwise it returns $\perp$.

Suppose $\widehat{c} = \Pi.\mathsf{Encode}(pk, m; R) \oplus e$ with $|e| \le \delta_{rec} N$. Write $e = e_{\mathsf{seed}} \| e_{\mathsf{pay}}$, where $e_{\mathsf{pay}} \in \{0,1\}^{\ell n_1}$ is the error on the payload block. Then we have $|e_{\mathsf{pay}}| \le |e| \le \delta_{rec} N$. Hence, if $\delta_{rec} N \le \alpha \ell n_1$, then $\widehat{y} \oplus \mathsf{PRG}(s) = \mathsf{Enc}_2(m) \oplus e_{\mathsf{pay}}$ lies within the decoding radius of ECC_2, and therefore

$$m^* = \mathsf{Dec}_2\big(\widehat{y} \oplus \mathsf{PRG}(s)\big) = m.$$

The re-encoding step then reconstructs the original honest codeword,

$$c^* = \Pi.\mathsf{Encode}(pk, m; R),$$

so the final check accepts. Thus

$$\mathsf{Rec}(pk, \Pi.\mathsf{Encode}(pk, m; R) \oplus e, R) = m$$

for every message m, every randomness R, and every error vector e of weight at most $\delta_{rec} N$.

Therefore, the construction admits recovery from randomness for every $\delta_{rec} \le \frac{\alpha}{2}$. In particular, this covers the robustness threshold $\alpha\beta/2$ proved for the construction in [AAC+25, Theorem 12].