Adaptively-Secure Unbounded Optimal DBE from Lattice Assumptions

Utkarsh Sahai[1], Arijit Saha[2], Ramprasad Sarkar[1], Mriganka Mandal[1]

[1] Cryptology and Security Research Unit, Indian Statistical Institute, Kolkata 700108
sahai.utkarsh13@gmail.com,{rpsarkar_p,mriganka}@isical.ac.in

[2] Department of Mathematics, Indian Institute of Technology, Jodhpur 342037
arijit.saha210999@gmail.com

September 5, 2025

Abstract. Distributed Broadcast Encryption (DBE) is a registration-based cryptographic paradigm in which users generate their own public/secret keys and register their public keys on a public bulletin board. Any sender can encrypt messages for any subset of registered users, and only intended recipients can decrypt, thereby enabling scalable, trustless, and provably secure communication. While Wee *et al.* (CRYPTO 2025) constructed the first unbounded optimal DBE from the succinct LWE assumption, their construction achieves only selective security. Until now, it was unknown how to achieve adaptive security for unbounded optimal DBE under standard lattice assumptions without relying on heavy primitives such as indistinguishability obfuscation or witness encryption. In this work, we present the first adaptively secure unbounded optimal DBE schemes from falsifiable lattice assumptions. Specifically, for the security parameter λ and number of users N, we achieve the following:

- A semi-statically secure DBE in the plain model for an arbitrary polynomial number of users, where the sizes of public parameters, user public/secret keys and ciphertext are all optimal (*i.e.*, have size $\mathsf{poly}(\lambda, \log N)$), based on the falsifiable $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption.
- An adaptively-secure DBE in the random oracle model supporting an arbitrary polynomial number of users, with optimal public parameters, user public/secret keys and ciphertext sizes, again under $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption.
- An adaptively-secure DBE in the plain model supporting a priori-maximum polynomially many users under the $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption. Our construction achieves optimal sizes for both the user public/secret keys and the ciphertext, whereas the public parameters grow linearly with the number of users (*i.e.*, have size $N \cdot \mathsf{poly}(\lambda, \log N)$).

Contents

1 Introduction . . . 3
1.1 Our Results . . . 5
1.2 Additional Related Works . . . 6
2 Technical Overview . . . 7
2.1 Towards Unbounded DBE . . . 9
2.2 Semi-Static Security from Succinct LWE . . . 12
2.3 Transparent Setup . . . 14
2.4 From Semi-Static to Adaptive Security . . . 14
3 Preliminaries . . . 16
3.1 Lattices Background . . . 17
3.2 Matrix Commitment . . . 21
3.3 Publicly-Sampleable Projective PRGs . . . 21
3.4 Distributed Broadcast Encryption . . . 23
4 Partitionable Matrix Commitment . . . 25
5 Semi-Statically Secure DBE in Plain Model . . . 29
5.1 Construction of Unbounded Optimal DBE from Succinct LWE . . . 29
5.2 Proof of Correctness and Succinctness . . . 30
5.3 Proof of Security . . . 34
5.4 Parameter Instantiation . . . 40
6 Transparent Setup via Decomposed LWE . . . 40
7 Adaptively-Secure DBE in Random Oracle Model . . . 42
7.1 Construction of Unbounded Optimal DBE from Succinct LWE . . . 42
7.2 Proof of Correctness and Succinctness . . . 44
7.3 Proof of Security . . . 46
7.4 Parameter Instantiation . . . 46
8 Adaptively-Secure DBE in Plain Model . . . 47
8.1 Construction of Almost Optimal DBE from Succinct LWE . . . 47
8.2 Proof of Correctness and Succinctness . . . 49
8.3 Proof of Security . . . 50
8.4 Parameter Instantiation . . . 51

1 Introduction

Designing advanced cryptographic primitives that do not rely on any trusted authority has become increasingly crucial due to the need for decentralization and provable security under minimal assumptions. In this context, we focus on Broadcast Encryption (BE) [FN93], an advanced cryptographic paradigm that has long relied on trusted authorities. BE enables a sender to encrypt messages for a target set $\mathcal{S}$ such that only members of $\mathcal{S}$ can decrypt, with the primary goal of minimizing the size of the ciphertext, even in systems with an arbitrarily large number of users. However, traditional BE schemes depend on a central authority to issue decryption keys to all recipients. This dependency introduces a severe key escrow problem: the compromise of the authority results in a single point of failure, allowing the recovery of all user keys and thus breaking confidentiality.

Distributed Broadcast Encryption. To overcome the aforementioned limitation, Distributed Broadcast Encryption (DBE) [WQZDF10, BZ14, KMW23, FWW23, CW24, CHW25, WW25] removes the need for a central authority. In DBE, users independently generate their public/secret key pairs and publish their public keys on a public bulletin board. Then anyone can encrypt a message to a set $\mathcal{S}$ using only the recipients' public keys, without relying on a trusted party. Existing DBE schemes have been realized under a range of cryptographic assumptions, including indistinguishability obfuscation [BZ14], witness encryption [FWW23], pairing-based approaches [WQZDF10, KMW23, GKPW24], and lattice-based designs [CW24, CHW25, WW25]. These existing DBE schemes can be further classified based on the following properties.

- **Trusted vs. Transparent Setup.** A fundamental classification of DBE schemes is based on the nature of the setup algorithm. Although the primary goal of a DBE scheme is to eliminate the need for a trusted key generator, some DBE schemes [KMW23, CW24, CHW25] still rely on a (one-time) *trusted setup*, where a designated authority initializes the system and publishes the system public parameters, raising concerns about hidden trapdoors or parameter subversion. In contrast, a *transparent setup* dispenses with any trusted authority and instead relies only on publicly verifiable randomness, thereby strengthening trust and decentralization. The schemes of [WQZDF10, BZ14, FWW23] achieve a transparent setup. The main construction of [WW25] requires a (one-time) trusted setup but can be converted to a transparent setup using the recently introduced Decomposed-LWE assumption [AMR25].
- **Unbounded vs. Bounded.** Another distinction among DBE schemes is according to the number of users that the scheme can support. In an *unbounded* DBE, the system can accommodate an arbitrary polynomial number of users. On the other hand, a *bounded* setup restricts to a *priori-maximum* polynomial number of system users. Unbounded DBE schemes include [BZ14, FWW23, WW25], while [WQZDF10, KMW23, CW24, CHW25] adopted only the bounded settings.

- **Optimality.** A DBE scheme is said to be *optimal* if the sizes of all the scheme parameters, such as public parameters, user public/secret keys, and ciphertext, scale polylogarithmically in the total number of system users N. The DBE constructions of [BZ14, FWW23, WW25] achieved the desired optimality conditions. While [WQZDF10] incurs a linear growth in both system public parameters and user secret keys, but quadratic growth in user public keys, the works of [KMW23, CW24, CHW25] feature linear-size user public keys with quadratic-size system public parameters.
- **Security.** The existing DBE schemes can also be classified according to the strength of their security guarantees. In selective security, the adversary must commit to the challenge set of users before seeing the system parameters. In contrast, semi-static security allows the adversary to commit to a superset $\mathcal{S}^*$ of users before seeing the public parameters and then choose the actual challenge set $\mathcal{S} \subseteq \mathcal{S}^*$ later. The strongest notion is adaptive security, where the adversary is allowed to adaptively corrupt users of its choice and commit to the challenge set at any point during the security game.

  A significant advantage of the semi-statically secure DBE scheme is that generic compilers can transform any semi-statically secure scheme into an adaptively-secure one with minimal overhead: either in the Random Oracle Model (ROM) [GW09, KMW23], or in the plain model using publicly sampleable Projective Pseudorandom Generators (PRGs) [HWW25]. Note that the works of [WQZDF10, CW24, WW25] achieve only selective security and hence cannot be upgraded to adaptive security using the aforementioned generic compilers. In contrast, [BZ14, FWW23, KMW23] achieve semi-static security and can be elevated to adaptive security in both models. The DBE scheme of [CHW25] achieves semi-static security in the ROM and can be upgraded to adaptive security by adopting the compilers of [GW09, KMW23], whereas the plain-model compiler [HWW25] does not provide any additional benefits, as the scheme is inherently tied to the random oracles.
- **Underlying Assumptions.** Lastly, DBE constructions can also be distinguished by the cryptographic assumptions they rely on, which often involve trade-offs between efficiency, scalability, and conceptual complexity. Some schemes build upon heavyweight primitives such as Witness Encryption (WE) or Indistinguishability Obfuscation (iO), while others rely on more standard assumptions at the cost of larger keys or less scalable setups. For instance, the work of [BZ14] is based on iO, whereas the construction of [FWW23] employs WE. Given the looming threat of quantum adversaries, lattice-based assumptions are generally considered more attractive. Although WE can, in principle, be instantiated from lattice-based assumptions [VWW22], these instantiations typically rely on the private-coin variant of the *non-falsifiable evasive* LWE assumption [Wee22, Tsa22], whose security remains uncertain due to heuristic obfuscation-based counterexamples [VWW22] and more recent evidence [AMYY25]. On the other hand, several DBE schemes [CW24, CHW25, WW25] rely on the succinct LWE assumption [Wee24], which is regarded as a standard falsifiable lattice-based assump-

tion, free from obfuscation dependencies. Moreover, a variant of [WW25] can also be instantiated under the recently proposed decomposed LWE assumption [AMR25].

Taken together, these classifications and trade-offs across setup models, scalability, optimality, security notions, and underlying assumptions intrinsically lead to the following fundamental question regarding the actual feasibility and robustness of DBE constructions:

Can we construct an adaptively-secure unbounded optimal DBE from falsifiable lattice assumptions under a transparent setup?

1.1 Our Results

In this work, we answer the above question affirmatively and, to the best of our knowledge, present the first adaptively-secure unbounded optimal DBE scheme based on falsifiable lattice assumptions, specifically the ℓ-succinct LWE assumption [Wee24]. Informally, the ℓ-succinct LWE assumption states that $(\mathbf{D}, \mathbf{sD} + \mathbf{e})$ is pseudorandom, even given a trapdoor $\mathbf{T}$ for the related matrix $[\mathbf{I}_\ell \otimes \mathbf{D} \mid \mathbf{W}]$, where $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{\ell n\times m}$, and χ is the error distribution. Let λ be the security parameter and N denote the total number of users. We summarize our results as follows:

We construct a DBE scheme that achieves semi-static security based on the succinct LWE assumption in the plain model. Our scheme supports an arbitrary polynomial number of users, and all system parameters, including public parameters, users' public/secret key pairs, and ciphertext, are of size $\mathsf{poly}(\lambda, \log N)$, thus making the scheme optimal. This overcomes the security and efficiency limitations of previous lattice-based constructions [CW24, CHW25, WW25]. We discuss this in detail in Section 5. We then, in Section 6, also present an alternative variant of our semi-statically secure DBE, where the public parameters are sampled from a transparent setup with no more than polynomial blow-up in parameter sizes, thereby improving upon [CW24, CHW25].

We construct a DBE scheme that achieves the strongest notion of security, namely adaptive security. To achieve this, we transform the semi-statically secure DBE from Section 5 into an adaptively-secure scheme in the random oracle model, employing techniques from [GW09, KMW23]. This transformation preserves the support for an arbitrary polynomial number of users and ensures that all system parameters, including public parameters, keys, and ciphertext, remain of optimal polylogarithmic size. The detailed transformation is in Section 7.

We further upgrade our semi-statically secure DBE to achieve adaptive security in the plain model by leveraging the recently introduced publicly sampleable Projective PRGs [HWW25]. While this approach requires fixing a priori-maximum number of users, it supports all users up to this bound with short user public/secret keys, as well as ciphertexts. The public parameters, however, scale linearly with the number of users. The details are presented in Section 8.

These results are summarized and contextualized via a comparative analysis in Table 1.

Scheme	Assumption	Parameters				Transparent Setup	Post Quantum	Unboundedness	Security Model
		|PP|	|CT|	|pk|	|sk|				
[WQZDF10]	Bilinear Diffie-Hellman Exponent	N	1	N^2	N	✓	✗	✗	selective
[BZ14]	$i\mathsf{O}$ + One way functions	–	1	1	1	✓	✗	✓	semi-static
[BZ14] + [GW09]		1	1	1	1	✓	✗	✓	adaptive†
[FWW23]	Witness Encryption + LWE	1	1	1	1	✓	✓	✓	semi-static
[FWW23] + [GW09]		1	1	1	1	✓	✓	✓	adaptive†
[FWW23] + [HWW25]		N	1	1	1	✓	✓	✗	adaptive
[KMW23]–I	Bilinear Diffie-Hellman Exponent	N	1	N	1	✗	✗	✗	semi-static
[KMW23]–I + [GW09]		N	1	N	1	✗	✗	✗	adaptive†
[KMW23]–I + [HWW25]		N	1	N	1	✗	✗	✗	adaptive
[KMW23]–II	k-Lin ($k \geq 1$)	N^2	1	N	1	✗	✗	✗	adaptive
[CW24]	poly(λ, N)-succinct LWE	N^2	1	N	1	✗	✓	✗	selective
[CHW25]	poly(λ, N)-succinct LWE	N^2	1	N	1	✗	✓	✗	semi-static†
[CHW25] + [GW09]		N^2	1	N	1	✗	✓	✗	adaptive†
[WW25]	poly$(\lambda, \log N)$-succinct LWE	1	1	1	1	✗‡	✓	✓	selective
Scheme–I [**Sec. 5**]	poly$(\lambda, \log N)$-succinct LWE	1	1	1	1	✗‡	✓	✓	semi-static
Scheme–II [**Sec. 7**]		1	1	1	1	✗‡	✓	✓	adaptive†
Scheme–III [**Sec. 8**]		N	1	1	1	✗‡	✓	✗	adaptive

‡A variant of these constructions can be obtained with a transparent setup using the Decomposed LWE assumption [AMR25]. For more details, refer to Section 6.

†The security of these schemes relies on the random oracle model.

Table 1: Comparison of prior DBE schemes and our constructions, covering assumptions, parameter sizes, setup type, post-quantum security, unboundedness, and security model. The Parameters columns report sizes of public parameters (|PP|), ciphertext (|CT|), public key (|pk|), and secret key (|sk|), all as a function of the number of users N. For clarity, poly$(\lambda, \log N)$ factors are suppressed, with λ being the security parameter. Indistinguishability obfuscation is denoted by $i\mathsf{O}$ [BGI+01]. Each scheme indicates whether the public parameters support a transparent setup, whether it is plausibly post-quantum secure, and the security model achieved, which is selective, semi-static, or adaptive.

1.2 Additional Related Works

Decentralized Broadcast Encryption. A fundamental method was introduced to eliminate key escrow in broadcast encryption in [PPS12], presenting the concept of decentralized broadcast encryption. In this scheme, users must participate in an interactive Multi-Party Computation (MPC) to create and update secret keys after adding a new user. This method requires collaboration and vigilance among users, which can be restrictive in its application. Conversely, DBE provides a more scalable and adaptable option: users autonomously produce their public and private keys and only disseminate their public keys in a public directory (or a public bulletin board). The absence of coordination simplifies the deployment and maintenance of the system. However, another issue with

DBE is its dependence on a slot-based framework, where each user is assigned a unique slot, and encryption is restricted to groups of users occupying separate slots.

Flexible Broadcast Encryption. To solve this slotting issue, [FWW23] proposed the notion of Flexible Broadcast Encryption (FBE), which eliminates the slot limitation and allows encryption for any set of users. Their technique employs witness encryption with function-binding hash functions. Subsequently, [GLWW23] proposed a generic compiler capable of converting any DBE scheme into an FBE through combinatorial methods while maintaining the efficiency of the system. Additionally, [GKPW24] proposed a direct implementation of FBE from pairing-based assumptions.

Registration-Based Broadcast Encryption. All these concepts of decentralized schemes align with a comprehensive cryptographic framework termed *registration-based* cryptography, as proposed in [GHMR18]. The principal objective here is to remove the dependency on a trusted and reliable central authority in various systems, such as Identity-Based Encryption (IBE) [GHMR18, GHM+19, GV20, CES21, GKMR23, DKL+23, FKP23], Attribute-Based Encryption (ABE) [HLWW23, FWW23, ZZGQ23, GLWW24, AT24, CHW25, WW25], Functional Encryption (FE) [FFM+23, DPY25], and traitor tracing [BLM+24]. In registration-based cryptography, users generate their own public/secret key pairs and publish their public keys on a public bulletin board, while a transparent curator compiles them into concise public parameters. In particular, [FWW23] demonstrated that any Registered Attribute-Based Encryption (RABE) scheme with a single attribute and an always-accept policy can be transformed into a DBE scheme with succinct ciphertexts. Numerous RABE constructions have since been proposed under various cryptographic assumptions, including indistinguishability obfuscation [HLWW23], witness encryption [FWW23], pairing-based techniques [ZZGQ23, AT24, GKPW24], and lattice assumptions [CHW25, WW25].

2 Technical Overview

This section presents a high-level overview of our adaptively-secure optimal DBE scheme from the ℓ-succinct LWE [Wee24] to support an unbounded number of users. We present a high-level pictorial description of our results in Fig. 1.

Notations. We first summarize some of the notations used throughout this paper. We use λ to represent the underlying security parameter. Row vectors are denoted in bold lowercase (*e.g.*, $\mathbf{v}$), and matrices in bold uppercase (*e.g.*, $\mathbf{V}$). The notation $\|\mathbf{v}\|$ (resp., $\|\mathbf{V}\|$) denotes the maximum absolute value among the entries of $\mathbf{v}$ (resp., $\mathbf{V}$). For sampling, $y \leftarrow \mathcal{Y}$ denotes a sample drawn from a distribution $\mathcal{Y}$, and $y \xleftarrow{\$} \mathcal{X}$ denotes an element sampled uniformly at random from a set $\mathcal{X}$. We write $\approx_s$ and $\approx_c$ to denote "statistically close" and "computationally indistinguishable," respectively. The (centered) discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter χ is denoted by $\mathcal{D}_{\mathbb{Z}^m,\chi}$. Given $\mathbf{Y} \in \mathbb{Z}_p^{n \times n'}$ and $\sigma > 0$,

we let $\mathbf{D}^{-1}(\mathbf{Y}, \sigma)$ denote the distribution of a matrix $\mathbf{X}$ sampled from $\mathcal{D}_{\mathbb{Z}^{m\times n'},\sigma}$ conditioned on $\mathbf{DX} = \mathbf{Y} \bmod p$. The standard gadget matrix is denoted by $\mathbf{G}$. We use $\mathcal{O}(\cdot)$ for the asymptotic order notation, and $\mathsf{poly}(\cdot)$ and $\mathsf{negl}(\cdot)$ to denote a polynomial function and a negligible function, respectively. Finally, we use *curly underlines* to highlight terms with added noise. For instance, $\underset{\sim}{\mathbf{sD}}$ denotes $\mathbf{sD} + \mathbf{e}$, where $\mathbf{e}$ is a short noise vector.

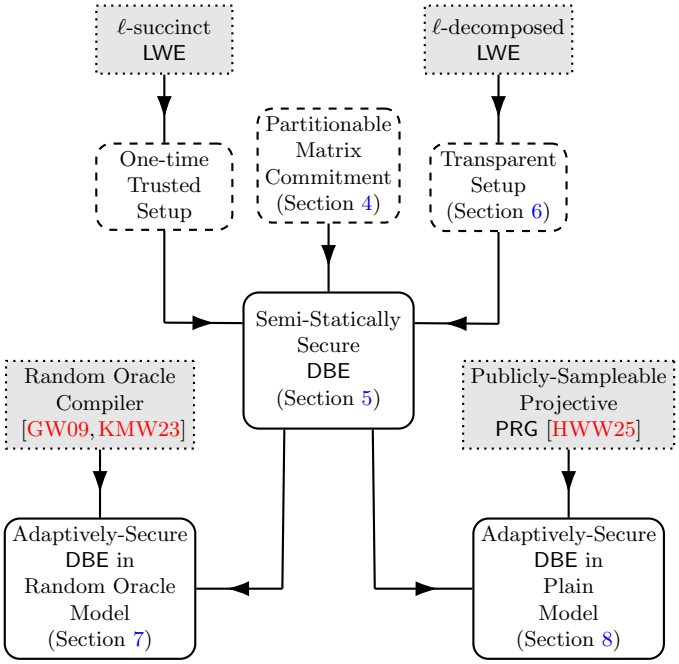


Fig. 1: High-level Pictorial Description of Our Results

ℓ-succinct LWE [Wee24]. Let n, m, p be the lattice parameters with $m = \mathcal{O}(n \log p)$. Sample $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{\ell n\times m}$, and a (Gaussian) matrix $\mathbf{T}$ such that $[\mathbf{I}_\ell \otimes \mathbf{D} \mid \mathbf{W}]\,\mathbf{T} = \mathbf{I}_\ell \otimes \mathbf{G}$, where $\mathbf{I}_\ell$ is the identity matrix of dimension ℓ. The ℓ-succinct LWE assumption asserts that

$$\left(\mathbf{D}, \underset{\sim}{\mathbf{sD}}, \mathbf{W}, \mathbf{T}\right) \approx_c (\mathbf{D}, \mathbf{u}, \mathbf{W}, \mathbf{T}),$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^m$. We refer to $(\mathbf{D}, \mathbf{W}, \mathbf{T})$, whose size is $\mathcal{O}(\ell^2)$, as the public parameters of a succinct LWE instance of dimension ℓ.

Matrix Commitment [Wee25, WW25]. Given a succinct LWE instance $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$ of dimension $2m^2$ and any arbitrary matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times L}$, we have an efficient and deterministic algorithm to compute a commitment $\mathbf{C} \in \mathbb{Z}_p^{n\times m}$ and

a low-norm opening $\mathbf{Z} \in \mathbb{Z}_p^{m \times L}$ such that

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z} \tag{1}$$

where, $\mathbf{V}$ is a fixed low-norm matrix also in $\mathbb{Z}_p^{m \times L}$ and is publicly derived from $\mathsf{pp}_{\mathsf{com}}$ and the width L.

2.1 Towards Unbounded DBE

In DBE schemes, each public key is indexed by some $i \in \mathbb{N}$, allowing encryption to any subset of users as long as the indices are distinct. Setting the index space as $N = 2^\lambda$ enables the scheme to naturally accommodate an arbitrary (polynomially unbounded) number of users by interpreting each index as an identity. To achieve scalability in this setting, the following two technical requirements must be necessarily addressed [WW25].

- **Compactness:** The size of all the parameters of the scheme, such as the public parameters (PP), public keys (pk) and secret keys (sk) of each user, and ciphertexts (CT), should be bounded by $\mathsf{poly}(\lambda, \log N)$.
- **Efficiency:** Key generation algorithm should not run in time greater than $\mathsf{poly}(\lambda, \log N)$, while the encryption and decryption algorithms should be bounded by $|\mathcal{S}| \cdot \mathsf{poly}(\lambda, \log N)$ time, where $\mathcal{S}$ denotes the broadcast set.

Building on the above requirements, we now revisit the notion of an unbounded DBE scheme [BZ14, FWW23, WW25]:

– **Setup**: Given the total number of system users N and the security parameter λ, any party executes (one-time) trusted setup to sample the public parameters PP. We assume that PP contains the information of N and 1^λ.
– **Key-Generation**: As each user is associated with a distinct index $i \in [N]$, given the public parameters PP, the user i generates a public-secret key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$. The public key pk_i is then posted on a public bulletin board.
– **Encryption**: Given the public parameters PP, a set of public keys $\{\mathsf{pk}_i\}_{i \in \mathcal{S}}$, and a message μ, the encryption algorithm outputs the ciphertext CT.
– **Decryption**: Taking as input the ciphertext CT, the public parameters PP, the associated set of public keys $\{\mathsf{pk}_i\}_{i \in \mathcal{S}}$, and the secret key sk_j for some users $j \in \mathcal{S}$, the decryptor outputs the message μ.

Starting Point. As a warm-up to our actual scheme, in the following, we first describe a scheme supporting N users that satisfies efficiency, but sacrifices compactness.

– **Setup**: Given the total number of users N and the security parameter λ, the public parameters are set as $\mathsf{PP} = (N, \mathbf{B}, \mathbf{L})$, where $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$ and $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n \times (\lceil \log N \rceil + 1)}$.

- **Key-Generation**: As each user is associated with a distinct index $i \in [N]$, the user $i \in [N]$ samples $\mathbf{h}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m$ to compute $\mathbf{d}_i^\top = \mathbf{L}\mathbf{h}_i^\top + \mathbf{B}\boldsymbol{\gamma}_i^\top$, where $\boldsymbol{\gamma}_i \in \{0,1\}^{(\lceil \log N \rceil + 1)}$ is the canonical binary representation of i. Then, the user sets the public-secret key pair as $(\mathsf{pk}_i, \mathsf{sk}_i) = (\mathbf{d}_i, \mathbf{h}_i)$. The public key is then posted on a public bulletin board.
- **Encryption**: Let $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ be the (ordered) broadcast set and $\mu \in \{0,1\}$ be the message to be encrypted. Samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$ and sets the matrix $\mathbf{M} = \left[\mathbf{d}_{i_1}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_1}^\top \mid \mathbf{d}_{i_2}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_2}^\top \mid \cdots \mid \mathbf{d}_{i_\varsigma}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_\varsigma}^\top\right] \in \mathbb{Z}_p^{n \times \varsigma}$. Finally, output the ciphertext as

$$\mathsf{CT} = \left(\overbrace{\underbrace{\mathbf{s}\mathbf{L}}_{} + \mu\mathbf{g}}^{\mathbf{c}_1}, \overbrace{\underbrace{\mathbf{s}\mathbf{M}}_{}}^{\mathbf{c}_2} \right),$$

  where $\mathbf{g}$ is the gadget vector. It is easy to verify that the size of the ciphertext grows linearly with $\mathcal{S}$.
- **Decryption**: Takes as input any user $j \in \mathcal{S}$. Based on the structure of $\mathcal{S}$, there exists $k \in [\varsigma]$ such that $j = i_k$. Now, we have

$$\begin{aligned}
\mathbf{c}_2 &= \underbrace{\mathbf{s}\mathbf{M}}_{} \\
&= \left[\underbrace{\mathbf{s}(\mathbf{d}_{i_1}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_1}^\top)}_{} \mid \underbrace{\mathbf{s}(\mathbf{d}_{i_2}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_2}^\top)}_{} \mid \cdots \mid \underbrace{\mathbf{s}(\mathbf{d}_{i_\varsigma}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_\varsigma}^\top)}_{} \right] \\
&= \left[\underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_1}^\top}_{} \mid \underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_2}^\top}_{} \mid \cdots \mid \underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_\varsigma}^\top}_{} \right].
\end{aligned}$$

  To decrypt, the user $j = i_k$ parses $\underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top}_{}$ from $\mathbf{c}_2$ and computes

$$\begin{aligned}
\mathbf{c}_1\mathbf{h}_{i_k}^\top - \underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top}_{} &= (\underbrace{\mathbf{s}\mathbf{L}}_{} + \mu\mathbf{g})\mathbf{h}_{i_k}^\top - \underbrace{\mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top}_{} \\
&\approx \underbrace{\mu\mathbf{g}\mathbf{h}_{i_k}^\top}_{}.
\end{aligned}$$

  The user rounds the above to recover the message.

As this is a starting point for us, we are primarily interested in the correctness of the above approach, rather than its security. It is interesting to see that if the noise is small and the quantity $\mathbf{g}\mathbf{h}_{i_k}^\top$ is statistically close to a uniform element over $\mathbb{Z}_p$, then the correctness follows easily. In addition, the size of the public parameters and the public-secret key pair is $\mathsf{poly}(\lambda, \log N)$. However, as mentioned before, the size of the ciphertext grows linearly with $\mathcal{S}$. In case $\mathcal{S}$ is super-poly, the construction is rendered useless as it does not satisfy the compactness. Therefore, we need to reduce the size of the ciphertext while retaining efficiency.

First Modification: Attempting Matrix Commitment. In the approach above, we try to reduce the size of the ciphertext by introducing matrix commitment, where we will try to generate a matrix $\mathbf{C}$ committing to the matrix $\mathbf{M}$. To accommodate this, we modify our approach as follows:

- **Setup**: We add a succinct LWE instance $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$ of dimension $2m^2$ in the public parameters, making the overall public parameters to be $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}}, \mathbf{B}, \mathbf{L})$.
- **Key-Generation**: The key generation algorithm remains the same as before.
- **Encryption**: Let $\mathbf{C} \in \mathbb{Z}_p^{n \times m}$ be the commitment to $\mathbf{M}$ with respect to $\mathbf{D}$ and further let $\mathbf{Z} \in \mathbb{Z}_p^{m \times \varsigma}$ be the respective opening. Hence, for a verification matrix $\mathbf{V} \in \mathbb{Z}_p^{m \times \varsigma}$, from Equation 1, we have

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}.$$

  Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, and set the final ciphertext as

$$\mathsf{CT} = \left(\overbrace{\underbrace{\mathbf{sD}}_{\mathbf{c}_0}}, \overbrace{\underbrace{\mathbf{sL} + \mu\mathbf{g}}_{\mathbf{c}_1}}, \overbrace{\underbrace{\mathbf{sC}}_{\mathbf{c}_2}} \right).$$

  Here, we have reduced the size of the ciphertext, which is now sublinear.
- **Decryption**: Given the public keys of all users in the broadcast set $\mathcal{S}$, it is easy to reconstruct $\mathbf{M}$, and hence, can easily sample the corresponding opening $\mathbf{Z}$ for $\mathbf{M}$. The verification matrix $\mathbf{V}$ can also be publicly derived from $\mathsf{pp}_{\mathsf{com}}$ and the width ς. To decrypt, the user first computes

$$\begin{aligned} \mathbf{c}_2 \cdot \mathbf{V} + \mathbf{c}_0 \cdot \mathbf{Z} &= \mathbf{sC} \cdot \mathbf{V} + \mathbf{sD} \cdot \mathbf{Z} \\ &= \mathbf{s}\,(\mathbf{C} \cdot \mathbf{V} + \mathbf{D} \cdot \mathbf{Z}) \\ &= \mathbf{sM}. \end{aligned}$$

  The user now continues in the same way as before to recover the message.

Easier Decryption with Partitionable Matrix Commitment. As observed in the above solutions, the user $j = i_k$ only needs to access the k^{th} component of the vector $\mathbf{sM}$ for decryption. However, to reach this stage, the decryptor needs to perform the matrix multiplications over large matrices $\mathbf{V}$ and $\mathbf{Z}$ (in the case $\mathcal{S}$ is superpoly). We can further reduce the multiplication cost by utilizing the partitionable structure of the matrix $\mathbf{M}$. Note that we are not introducing any changes to the core Matrix Commitment algorithm. Instead of focusing on using the whole of $\mathbf{V}$ and $\mathbf{Z}$, we will focus only on the k^{th} column of each. Therefore, in our case, Equation 1 reduces to

$$\mathbf{C} \cdot \mathbf{V}_k = \mathbf{M}_k - \mathbf{D} \cdot \mathbf{Z}_k, \tag{2}$$

where $\mathbf{V} = [\mathbf{V}_1 \mid \mathbf{V}_2 \mid \cdots \mid \mathbf{V}_\varsigma]$, $\mathbf{Z} = [\mathbf{Z}_1 \mid \mathbf{Z}_2 \mid \cdots \mid \mathbf{Z}_\varsigma]$, and $\mathbf{M}_k = \left[\mathbf{d}_{i_k}^\top - \mathbf{B}\boldsymbol{\gamma}_i^\top\right]$. We discuss the correctness of Equation 2 in detail in Section 4.

With this, for decryption, the user $j = i_k$ now first parses the k^{th} column of both $\mathbf{V}$ and $\mathbf{Z}$, say $\mathbf{V}_k$ and $\mathbf{Z}_k$ respectively. Then, perform the following computations:

$$\begin{aligned}
\mathbf{c}_1\mathbf{h}_{i_k}^\top - (\mathbf{c}_2 \cdot \mathbf{V}_k + \mathbf{c}_0 \cdot \mathbf{Z}_k) &= (\underbrace{\mathbf{sL}} + \mu\mathbf{g})\mathbf{h}_{i_k}^\top - \left(\underbrace{\mathbf{sC}} \cdot \mathbf{V}_k + \underbrace{\mathbf{sD}} \cdot \mathbf{Z}_k\right) \\
&= (\underbrace{\mathbf{sL}} + \mu\mathbf{g})\mathbf{h}_{i_k}^\top - \left(\underbrace{\mathbf{s}(\mathbf{C} \cdot \mathbf{V}_k + \mathbf{D} \cdot \mathbf{Z}_k)}\right) \\
&= (\underbrace{\mathbf{sL}} + \mu\mathbf{g})\mathbf{h}_{i_k}^\top - \left(\underbrace{\mathbf{sLh}_{i_k}^\top}\right) \\
&= \underbrace{\mu\mathbf{gh}_{i_k}^\top}.
\end{aligned}$$

Following the construction so far, we observe that the public parameters, the public/secret keys of the users, and the size of the ciphertext are $\mathsf{poly}(\lambda, \log N)$, ensuring compactness. In addition, the running time for the key generation algorithm is $\mathsf{poly}(\lambda, \log N)$. Furthermore, the execution time of encryption and decryption is bounded by $|\mathcal{S}| \cdot \mathsf{poly}(\lambda, \log N)$, which makes construction efficient. Now, we need to argue for the security of this construction, which we discuss in the following.

2.2 Semi-Static Security from Succinct LWE

We first aim to achieve the semi-static security, where the adversary declares a target superset $\mathcal{S}^* \subseteq [N]$ of indices at the beginning of the security game, but sends the actual challenge set $\mathcal{S} \subseteq \mathcal{S}^*$ after seeing the public parameters and the public keys of the users. To achieve the semi-static security from succinct LWE, we need to show that we can simulate the challenge ciphertext given the succinct LWE instance $\left(\mathbf{D}, \underbrace{\mathbf{sD}}, \mathbf{W}, \mathbf{T}\right)$. In our construction, the first two components can easily be simulated from the given instance, as we set $\mathbf{c}_0 = \underbrace{\mathbf{sD}}$, and for $\mathbf{c}_1$, we set $\mathbf{L} = \mathbf{DK}_1$ for some $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m \times m}$. From the Leftover Hash Lemma, we get that $\mathbf{L}$ and $\mathbf{DK}_1$ are statistically indistinguishable. We finally set $\mathbf{c}_1 = \underbrace{\mathbf{sD}} \cdot \mathbf{K}_1 + \mu\mathbf{g}$. However, for now, we do not have any such direct result for the component $\mathbf{c}_2$. We need to program the challenge set $\mathcal{S}$ in some way in the final stage of the security analysis. To achieve this, we further modify the encryption and decryption algorithms as follows.

Programming the Challenge Set. In the semi-static security game, the adversary submits the challenge set $\mathcal{S}$ after observing both the public parameters and the public keys of all the users in the pre-declared target superset $\mathcal{S}^*$. For the security analysis, it is necessary to program this challenge set $\mathcal{S}$ into the challenge ciphertext. To achieve this, we slightly modify the aforementioned approach as follows.

– **Setup**: The setup algorithm is the same as before, and the public parameters are $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}}, \mathbf{B}, \mathbf{L})$.
– **Key-Generation**: The key generation algorithm also remains the same as before, where $(\mathsf{pk}_i, \mathsf{sk}_i) = (\mathbf{d}_i, \mathbf{h}_i)$ are sampled similarly as before.
– **Encryption**: Similarly to our previous approach, $\mathbf{C} \in \mathbb{Z}_p^{n \times m}$ is the commitment to $\mathbf{M}$ with respect to $\mathbf{D}$. Hence, for a verification matrix $\mathbf{V} \in \mathbb{Z}_p^{m \times \varsigma}$, from Equation 1, we obtain the following:

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}.$$

At last, sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$ and set $\mathbf{g}' = [\mathbf{g} \mid 0^*] \in \mathbb{Z}_p^{2m}$. The final ciphertext is

$$\mathsf{CT} = \left(\overbrace{\mathbf{sD}}^{\mathbf{c}_0},\ \overbrace{\mathbf{s}(\mathbf{L} \mid \mathbf{A}) + \mu \mathbf{g}'}^{\mathbf{c}_1},\ \overbrace{\mathbf{s}(\mathbf{C} + \mathbf{A})}^{\mathbf{c}_2} \right).$$

The matrix $\mathbf{A}$ will be used to program the challenge set in the security analysis. Also, observe that the size of the ciphertext increases only by a factor of $\mathcal{O}(m \log p)$, preserving the compactness.
– **Decryption**: For the decryption, the user $j = i_k$ first parses $\mathbf{V}_k$ and $\mathbf{Z}_k$, same as before, and performs the following computations:

$$\begin{aligned}
&\mathbf{c}_1 \begin{pmatrix} \mathbf{h}_{i_k}^\top \\ \mathbf{V}_k \end{pmatrix} - (\mathbf{c}_2 \cdot \mathbf{V}_k + \mathbf{c}_0 \cdot \mathbf{Z}_k) \\
&\qquad = (\mathbf{s}(\mathbf{L} \mid \mathbf{A}) + \mu \mathbf{g}') \begin{pmatrix} \mathbf{h}_{i_k}^\top \\ \mathbf{V}_k \end{pmatrix} - \left(\mathbf{s}(\mathbf{C} + \mathbf{A}) \cdot \mathbf{V}_k + \mathbf{sD} \cdot \mathbf{Z}_k \right) \\
&\qquad = \mathbf{s}(\mathbf{L} \mid \mathbf{A}) \begin{pmatrix} \mathbf{h}_{i_k}^\top \\ \mathbf{V}_k \end{pmatrix} + \mu \mathbf{g} \mathbf{h}_{i_k}^\top - \left(\mathbf{s}(\mathbf{C} \cdot \mathbf{V}_k + \mathbf{D} \cdot \mathbf{Z}_k) \right) - \mathbf{sA} \cdot \mathbf{V}_k \\
&\qquad = \mathbf{sLh}_{i_k}^\top + \mathbf{sA} \cdot \mathbf{V}_k + \mu \mathbf{g} \mathbf{h}_{i_k}^\top - \left(\mathbf{sLh}_{i_k}^\top \right) - \mathbf{sA} \cdot \mathbf{V}_k \\
&\qquad = \mu \mathbf{g} \mathbf{h}_{i_k}^\top.
\end{aligned}$$

We now revisit the challenge of simulating the ciphertext based on the succinct LWE instance. The reduction for the proof of semi-static security has the following high-level procedures:

- For all $i \in \mathcal{S}^*$, the challenger samples the public keys as $\mathbf{d}_i \xleftarrow{\$} \mathbb{Z}_p^n$. From Gaussian sampling with large width, we have that honestly generated public keys $\mathbf{d}_i$, where $\mathbf{d}_i^\top = \mathbf{L}\mathbf{h}_i^\top - \mathbf{B}\boldsymbol{\gamma}_i^\top$, are statistically indistinguishable from a uniform random vector over $\mathbb{Z}_p^n$.
- When setting the challenge ciphertext, the challenger embeds the challenge set into the third component $\mathbf{c}_2$ of the ciphertext by setting $\mathbf{A} = \mathbf{DK}_2 - \mathbf{C}$, where $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m \times m}$ and $\mathbf{C}$ represents the commitment to the matrix $\mathbf{M}$

constructed on the public keys in the challenge set. In addition, the component $\mathbf{c}_1$ is now set to $\underset{\sim}{\mathbf{sD}} \cdot \mathbf{K}_1 + \mu\mathbf{g}'$, where $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$. Again, from the leftover hash lemma, the distributions are statistically indistinguishable.

- Based on the above modifications, the challenge ciphertext can now be written as follows:

$$\mathsf{CT} = \left(\underset{\sim}{\mathbf{sD}},\ \underset{\sim}{\mathbf{s}(\mathbf{L} \mid \mathbf{A})} + \mu\mathbf{g}',\ \underset{\sim}{\mathbf{s}(\mathbf{C}+\mathbf{A})}\right) \approx_s \left(\underset{\sim}{\mathbf{sD}},\ \underset{\sim}{\mathbf{sD}} \cdot \mathbf{K}_1 + \mu\mathbf{g}',\ \underset{\sim}{\mathbf{sD}} \cdot \mathbf{K}_2\right),$$

  which can be easily simulated from $\underset{\sim}{\mathbf{sD}}$. From the $2m^2$-succinct LWE, $\underset{\sim}{\mathbf{sD}}$ is pseudorandom given the succinct LWE instance $(\mathbf{D}, \mathbf{W}, \mathbf{T})$ of dimension $2m^2$. Combining these results, we find that the challenge ciphertext behaves like pseudorandom and the semi-static security holds.

We informally summarize the description of construction and the extensive security analysis as a theorem as follows.

Theorem 2.1 (Informal). *Let $\lambda \in \mathbb{N}$ be the security parameter and let $N = 2^\lambda$ be a bound on the total number of users. Then, assuming the hardness of the $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption, there exists a semi-statically secure unbounded optimal DBE scheme in the plain model.*

We discuss this informal theorem in detail in Section 5.

2.3 Transparent Setup

In our construction, the setup phase follows the standard trapdoor-based commitment framework, where the system parameters include a matrix $\mathbf{D}$ with an associated Gaussian trapdoor $\mathbf{T}$. As a result, the default setup requires a (one-time) trusted authority. However, from the works of [Wee25], the matrix commitment can alternatively be instantiated using the decomposed-LWE assumption [AMR25]. Adopting this substitution, the public parameters can be sampled from a uniform random string, thereby yielding a fully transparent setup at the expense of *polynomially* larger parameters. Our security proof then relies on the decomposed-LWE assumption, in line with recent works of [WW25]. We discuss this substitution in detail in Section 6.

2.4 From Semi-Static to Adaptive Security

The proposed DBE scheme, as constructed till now, achieves semi-static security under the succinct LWE assumption. We do not directly construct an adaptively-secure DBE scheme from scratch. Instead, we use the techniques presented in [GW09, KMW23, HWW25] to systematically upgrade our semi-statically secure DBE scheme into an adaptive one. The template for building an adaptively-secure scheme $\prod^{\mathsf{Ad}}_{\mathsf{DBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ from a semi-statically secure one $\prod^{\mathsf{SS}}_{\mathsf{DBE}} = (\mathsf{SS.Setup}, \mathsf{SS.KeyGen}, \mathsf{SS.Enc}, \mathsf{SS.Dec})$ is described below.

– **Setup**: Given the security parameter λ and the total number of users N, sample $\mathsf{SS.PP} \leftarrow \mathsf{SS.Setup}\left(1^\lambda, 2N\right)$. Here, we assume that the underlying semi-statically secure DBE scheme supports $2N$ many users, and for ease of exposition, we index the set $[2N]$ using a pair $(i, \beta) \in [N] \times \{0,1\}$. The public parameters are set as $\mathsf{PP} = \mathsf{SS.PP}$.

– **Key-Generation**: The user, with index i, samples $\left(\mathsf{SS.pk}_{(i,0)}, \mathsf{SS.sk}_{(i,0)}\right) \leftarrow \mathsf{SS.KeyGen}\ (\mathsf{SS.PP},\ (i,0))$ and $\left(\mathsf{SS.pk}_{(i,1)}, \mathsf{SS.sk}_{(i,1)}\right) \leftarrow \mathsf{SS.KeyGen}\ (\mathsf{SS.PP},\ (i,1))$. It then samples a random bit $s_i \xleftarrow{\$} \{0,1\}$, and outputs the public-secret key pair $\mathsf{pk}_i = \left(\mathsf{SS.pk}_{(i,0)}, \mathsf{SS.pk}_{(i,1)}\right)$ and $\mathsf{sk}_i = \left(s_i, \mathsf{SS.sk}_{(i,s_i)}\right)$.

– **Encryption**: Taking as input the message $\mu \in \{0,1\}$, a set of indices $\mathcal{S} \subseteq [N]$, and the respective public keys $\left\{\mathsf{pk}_i = \left(\mathsf{SS.pk}_{(i,0)}, \mathsf{SS.pk}_{(i,1)}\right)\right\}_{i\in\mathcal{S}}$, the encryption algorithm performs the following steps:

1. For each $i \in \mathcal{S}$, sample $t_i \xleftarrow{\$} \{0,1\}$ to set $\mathcal{S}_0 = \{(i,t_i)\}_{i\in\mathcal{S}}$ and $\mathcal{S}_1 = \{(i,1-t_i)\}_{i\in\mathcal{S}}$.
2. Compute ciphertext components as follows:

$$\mathsf{SS.CT}_0 \leftarrow \mathsf{SS.Enc}\left(\mathsf{SS.PP}, \mathcal{S}_0, \left\{\left((i,t_i), \mathsf{SS.pk}_{(i,t_i)}\right)\right\}_{i\in\mathcal{S}}, \mu\right),$$

$$\mathsf{SS.CT}_1 \leftarrow \mathsf{SS.Enc}\left(\mathsf{SS.PP}, \mathcal{S}_1, \left\{\left((i,1-t_i), \mathsf{SS.pk}_{(i,1-t_i)}\right)\right\}_{i\in\mathcal{S}}, \mu\right).$$

The final ciphertext is $\mathsf{CT} = (\{t_i\}_{i\in\mathcal{S}},\ \mathsf{SS.CT}_0,\ \mathsf{SS.CT}_1)$, making the size of the ciphertext linear to the broadcast set $\mathcal{S}$.

– **Decryption**: To decrypt, the user j sets $b = t_j \oplus s_j$ and recovers the message as follows:

$$\mu = \mathsf{SS.Dec}\big(\mathsf{SS.PP}, \mathcal{S}_b, \left\{\left((i, t_i \oplus b), \mathsf{SS.pk}_{(i,t_i\oplus b)}\right)\right\}_{i\in\mathcal{S}}, \mathsf{SS.CT}_b, \left((j,s_j), \mathsf{SS.sk}_{(j,s_j)}\right)\big).$$

The correctness of the above construction directly follows from the correctness of the underlying semi-statically secure DBE scheme. On the other hand, adaptive security for up to N users follows directly from the works of [KMW23], as long as the underlying DBE scheme is semi-statically secure for up to $2N$ many users. Considering our semi-statically secure DBE supports an unbounded number of users, the correctness and adaptive security of the transformed construction are easily achievable and follow directly. Note that the efficiency of the scheme is also maintained. However, compactness is sacrificed because there is a bit string of size $|\mathcal{S}|$ included in the ciphertext. Hence, this transformation alone does not provide an unbounded optimal DBE scheme. Interestingly, we can still achieve the desired result through the following modifications.

Reclaiming Succinctness through Random Oracles [GW09, KMW23]**.** The problem with the above transformation, as already mentioned, is the linear

size of the ciphertext, which is due to the addition of the bit string $\{t_i\}_{i\in\mathcal{S}}$ in the final ciphertext. However, we have a solution to this problem, which enables us to achieve an adaptively-secure DBE scheme with truly succinct ciphertexts in the random oracle model, as follows.

Put a cryptographically secure hash function $\mathcal{H} : \{0,1\}^{\mathcal{O}(\lambda)} \times [N] \to \{0,1\}$ in public parameters. The encryption algorithm runs similarly to what was mentioned before. Instead of sampling $t_i \leftarrow \{0,1\}$, it samples $\mathbf{u} \xleftarrow{\$} \{0,1\}^{\mathcal{O}(\lambda)}$ and sets $t_i = \mathcal{H}(\mathbf{u}, i)$ for each user index $i \in \mathcal{S}$. Finally, in the ciphertext, the bit string $\{t_i\}_{i\in\mathcal{S}}$ is replaced with the vector $\mathbf{u}$. The recipient decrypts as before, except that it recovers $\{t_i\}_{i\in\mathcal{S}}$ from $\mathbf{u}$ using $\mathcal{H}$. With this modification, we reclaim both the compactness and efficiency by successfully constructing an unbounded optimal DBE from succinct LWE in the random oracle model. We can informally represent this modification as a theorem as follows.

Theorem 2.2 (Informal). *Let λ be the security parameter, and let $N = 2^\lambda$ be a bound on the number of users. Assuming the hardness of the poly(λ, $\log N$)-succinct LWE problem, there exists an unbounded optimal DBE scheme that achieves adaptive security in the random oracle model based on our semi-statically secure DBE scheme.*

We discuss this informal theorem in detail in Section 7.

Succinctness in Plain Model with Priori-Bounded Setting [HWW25]. In the above modification, we have integrated cryptographically secure hash functions as random oracles to achieve succinctness. However, from the work of [HWW25], we can also approach this problem of linear ciphertext in the plain model by using the LWE instance of publicly-sampleable projective pseudorandom generators (PRGs). The bottleneck we encounter with this approach is the inherent problem of linear-sized public parameters required for publicly-sampleable projective PRGs. We can still employ this technique where the total number of users supported by the system is a priori bounded, say $N = \mathsf{poly}(\lambda)$. We can informally represent this modification as a theorem as follows.

Theorem 2.3 (Informal). *Let λ be the security parameter and $N = N(\lambda)$ be any polynomial. Under the poly(λ, $\log N$)-succinct LWE assumption and a publicly sampleable projective pseudorandom generator from LWE satisfying sampling indistinguishability with succinct seed sizes, there exists an adaptively-secure almost optimal DBE scheme in the plain model based on our semi-statically secure DBE scheme.*

We discuss this informal theorem in detail in Section 8.

3 Preliminaries

This section outlines the essential preliminaries required to understand this work.

Tensor Product. The tensor (Kronecker) product for the matrices $\mathbf{A} = (a_{ij}) \in \mathbb{Z}^{n\times m}$ and $\mathbf{B} = (b_{ij}) \in \mathbb{Z}^{k\times \ell}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1m}\mathbf{B} \\ \vdots & \ldots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nm}\mathbf{B} \end{bmatrix} \in \mathbb{Z}^{nk \times m\ell}.$$

For matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, where $\mathbf{AC}$ and $\mathbf{BD}$ are well defined, the tensor product satisfies the following mixed-product property:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}).$$

We will state two useful corollaries of the mixed-product property:

- For a vector $\mathbf{x}$ and a matrix $\mathbf{A}$,

$$(\mathbf{x} \otimes \mathbf{I})\mathbf{A} = (\mathbf{x} \otimes \mathbf{I})(1 \otimes \mathbf{A}) = (\mathbf{x} \otimes \mathbf{A}).$$

- For a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}^{k \times \ell}$,

$$\mathbf{A} \otimes \mathbf{B} = (\mathbf{I}_n \otimes \mathbf{B})(\mathbf{A} \otimes \mathbf{I}_\ell) = (\mathbf{A} \otimes \mathbf{I}_k)(\mathbf{I}_m \otimes \mathbf{B}).$$

Vectorization. The vectorization of a matrix $\mathbf{A}$, denoted by $\mathsf{vec}(\mathbf{A})$, is the column vector formed by stacking the columns of the matrix $\mathbf{A}$ on top of one another. We have the following identity

$$\mathsf{vec}(\mathbf{ABC}) = \left(\mathbf{C}^\top \otimes \mathbf{A}\right) \mathsf{vec}(\mathbf{B}).$$

3.1 Lattices Background

Throughout this work, the discrete Gaussian distribution over $\mathbb{Z}$, characterized by a standard deviation χ, is represented using the notation $\mathcal{D}_{\mathbb{Z},\chi}$.

Learning with Errors (LWE). Given n, m, p, $\chi \in \mathbb{N}$, the $\mathsf{LWE}_{n,m,p,\chi}$ assumption states that

$$(\mathbf{A}, \boxed{\mathbf{sA} + \mathbf{e}}) \approx_c (\mathbf{A}, \mathbf{c}),$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}$ and $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^m$.

Gadget Matrices [MP12]**.** For $n, m, p \in \mathbb{N}$ with $m \geq n \cdot \lceil \log p \rceil$, a gadget matrix $\mathbf{G}_n \in \mathbb{Z}_p^{n \times m}$ is defined as $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}^\top$, where $\mathbf{g} = (1, 2, 4, \ldots, 2^{\lceil \log p \rceil - 1})$. It may be padded with zero columns to reach width m. We denote the standard deterministic entry-wise bit decomposition by $\mathbf{G}_n^{-1}(\cdot) : \mathbb{Z}_p^n \to \mathbb{Z}_p^m$. By construction for all $\mathbf{x} \in \mathbb{Z}_p^n$, we have that $\mathbf{G}_n \cdot \mathbf{G}_n^{-1}(\mathbf{x}^\top) = \mathbf{x}^\top \mod p$. We drop the subscript n when the context is clear.

ℓ-succinct LWE [Wee24]**.** Let λ be the security parameter and let n, m, p, χ be LWE parameters with $m \geq 2n \log p$. The $(\ell, \hat{m}, \sigma)$-succinct LWE assumption, where $m \leq \hat{m} \leq \ell m$, for the following distributions, asserts that:

$$(\mathbf{D}, \boxed{\mathbf{sD} + \mathbf{e}}, \mathbf{W}, \mathbf{T}) \approx_c (\mathbf{D}, \boxed{\mathbf{c}}, \mathbf{W}, \mathbf{T})$$

where, $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m$, $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^m$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{\ell n \times \hat{m}}$, and $\mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{D} \mid \mathbf{W}]^{-1}(\mathbf{I}_\ell \otimes \mathbf{G}, \sigma)$. Here, $\mathbf{T}$ is a trapdoor for the matrix $[\mathbf{I}_\ell \otimes \mathbf{D} \mid \mathbf{W}]$, generated using a gadget matrix $\mathbf{G}$ and with respect to width parameter σ.

We refer to this as (ℓ, σ)-succinct LWE assumption when $\hat{m} = m$, and simply as the ℓ-succinct LWE assumption when $\hat{m} = m$ and $\sigma = \mathsf{poly}(\lambda, \ell, m)$. The assumption is believed to be hard for the modulus-to-noise ratio $p/\chi \approx 2^{n^\epsilon}$, for some $0 < \epsilon < 1$.

ℓ-decomposed LWE [AMR25, Wee25]**.** Let λ be the security parameter and let n, m, p, χ be LWE parameters with $m \geq 2n \log p$. Let σ be the width parameter, then the ℓ-decomposed LWE assumption asserts that:

$$\left(\mathbf{D}, \boxed{\mathbf{s}\left(\mathbf{W}\left(\mathbf{I}_\ell \otimes \mathbf{R}\right) + \mathsf{vec}(\mathbf{I}_\ell)^\top \otimes \mathbf{G}\right) + \mathbf{e}}, \mathbf{W}, \mathbf{R}\right) \approx_c (\mathbf{D}, \boxed{\mathbf{c}}, \mathbf{W}, \mathbf{R})$$

where, $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell^2 m}$, $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^{\ell^2 m}$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{n \times \ell m}$, and $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m \times \ell m}$.

Lemma 3.1 (Gadget Trapdoor and Preimage Sampling [Ajt96, GPV08, CHKP12, MP12, AY20]**).** *Let n, m, p be parameters with $m \geq 3n \log p$. Then, there exist efficient algorithms* (TrapGen, SamplePre) *with the following syntax:*

- $(\mathbf{D}, \mathbf{T_D}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, p)$: *Given the inputs as lattice dimension n, width m, and the modulus p, the* TrapGen *algorithm outputs a matrix $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$ together with a trapdoor $\mathbf{T_D} \in \mathbb{Z}_p^{m \times m}$.*

- $\mathbf{X} \leftarrow \mathsf{SamplePre}(\mathbf{D}, \mathbf{T_D}, \mathbf{Y}, \sigma)$: *Given inputs a matrix $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$, a trapdoor $\mathbf{T_D} \in \mathbb{Z}_p^{m \times m}$, a target matrix $\mathbf{Y} \in \mathbb{Z}_p^{n \times n'}$ in the column space of $\mathbf{D}$, and a width parameter $\sigma > 0$, the* SamplePre *algorithm outputs a matrix $\mathbf{X} \in \mathbb{Z}_p^{m \times n'}$.*

The above algorithms satisfy the following properties:

Trapdoor Distribution: *If $(\mathbf{D}, \mathbf{T_D}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, p)$, then $\mathbf{D}$ is statistically indistinguishable from a uniform matrix over $\mathbb{Z}_p^{n \times m}$. Also, $\mathbf{DT_D} = \mathbf{G} \in \mathbb{Z}_p^{n \times m}$ and $\|\mathbf{T_D}\| = 1$.*

Preimage Sampling: *For all matrices $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$, $\mathbf{T_D} \in \mathbb{Z}_p^{m \times m}$, width parameters $\sigma > 0$, and all $\mathbf{Y} \in \mathbb{Z}_p^{n \times n'}$ in the image of $\mathbf{D}$, the output $\mathbf{X} \leftarrow \mathsf{SamplePre}(\mathbf{D}, \mathbf{T_D}, \mathbf{Y}, \sigma)$ satisfies $\mathbf{DX} = \mathbf{Y}$.*

Preimage Distribution: *For all matrices* $\mathbf{D} \in \mathbb{Z}_p^{n\times m}$ *and* $\mathbf{T_D} \in \mathbb{Z}_p^{m\times m}$ *such that* $\mathbf{DT_D} = \mathbf{G}$*, all width parameters* $\sigma \geq m\|\mathbf{T_D}\| \log n$*, and all targets* $\mathbf{Y} \in \mathbb{Z}_p^{n\times n'}$*, the distribution of the output from* $\mathsf{SamplePre}(\mathbf{D}, \mathbf{T_D}, \mathbf{Y}, \sigma)$ *is statistically indistinguishable from the output of* $\mathbf{D}^{-1}(\mathbf{Y}, \sigma)$*.*

Moreover, we have the property that if $\mathbf{D}^{-1}(\cdot,\cdot)$ is sampleable with respect to the trapdoor $\mathbf{T_D}$, then $[\mathbf{D} \mid \mathbf{Q}]^{-1}(\cdot,\cdot)$ and $[\mathbf{Q} \mid \mathbf{D}]^{-1}(\cdot,\cdot)$ are also sampleable with respect to $\mathbf{T_D}$ for any matrix $\mathbf{Q}$.

Lemma 3.2 (Generalized Leftover Hash Lemma [ABB10]). *Let* n*,* m*,* p *be integers such that* $m \geq 2n\log p$ *and* $p > 2$ *is prime. Then, for all fixed vectors* $\mathbf{e} \in \mathbb{Z}_p^m$ *and all* $k = poly(n)$*, the statistical distance between the following distributions is* $\mathsf{negl}(n)$*.*

$$\left\{\left(\mathbf{A}, \boxed{\mathbf{AR}}, \mathbf{e}^\top\mathbf{R}\right) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}, \mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times k}\right\}$$

and

$$\left\{\left(\mathbf{A}, \boxed{\mathbf{U}}, \mathbf{e}^\top\mathbf{R}\right) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m},\ \mathbf{U} \xleftarrow{\$} \mathbb{Z}_p^{n\times k},\ \mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times k}\right\}$$

Proposition 3.1 (Column Space of Random Matrix [GPV08]). *Let* n, m*,* p *be lattice parameters, where* p *is prime and* $m \geq 2n\log p$*. Then, for all but a* p^{-n} *fraction of matrices* $\mathbf{A} \in \mathbb{Z}_p^{n\times m}$*, the columns of* $\mathbf{A}$ *generate* $\mathbb{Z}_p^n$*.*

Lemma 3.3 (Gaussian Tail Bound [MP12]). *Let* n*,* m*,* p *be the lattice parameters, where* $m \geq 2n\log p$*. Sample a matrix* $\mathrm{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$*. Then, for all* $\chi > \log m$ *and all vectors* $\mathbf{t} \in \mathbb{Z}_p^n$ *in the column space of* $\mathbf{A}$*,*

$$\Pr\left[\|\mathbf{u}\| > \sqrt{m}\chi : \mathbf{u}^\top \leftarrow \mathbf{A}^{-1}(\mathbf{t}^\top, \chi)\right] \leq \mathcal{O}\left(2^{-m}\right).$$

For the particular case of discrete Gaussian over the integers and any $\lambda \in \mathbb{N}$*,*

$$\Pr\left[|x| > \sqrt{\lambda}\chi : x \leftarrow \mathcal{D}_{\mathbb{Z},\chi}\right] \leq 2^{-\lambda}.$$

Lemma 3.4 (Gaussian Samples [GPV08]). *Let* n, m, p, χ *be the lattice parameters such that* $\chi \geq \log m$*,* $m \geq 2n\log p$*, and* p *is a prime. Then, the statistical distance between the following two distributions is at most* $\mathsf{negl}(n)$*.*

$$\left\{(\mathbf{A}, \mathbf{x}, \boxed{\mathbf{Ax}^\top}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}, \mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m\right\}$$

and

$$\left\{(\mathbf{A}, \mathbf{x}, \boxed{\mathbf{t}^\top}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}, \mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^n, \mathbf{x}^\top \leftarrow \mathbf{A}^{-1}(\mathbf{t}^\top, \chi)\right\}$$

Lemma 3.5 (Marginal of Gaussian Preimages [WW23,CW24]). *Let n, m, p be lattice parameters where $m \geq 2n \log p$, and p is prime. Let $\ell, k = \mathsf{poly}(n, \log p)$. Then for all but a p^{-n} fraction of matrices $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$, all matrices $\mathbf{W} \in \mathbb{Z}_p^{n\ell \times k}$, all target vectors $\mathbf{t} \in \mathbb{Z}_p^{n\ell}$, and all width parameters $\sigma \geq 4 \log(\ell m)$, the statistical distance between the following distributions is at most $\mathsf{negl}(n)$:*

$$\left\{ \mathbf{x}^\top \ : \ \mathbf{x}^\top \leftarrow [\, \mathbf{I}_\ell \otimes \mathbf{D} \mid \mathbf{W} \,]^{-1} (\mathbf{t}^\top, \sigma) \right\}$$

and

$$\left\{ \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \end{bmatrix} \ : \ \begin{array}{c} \mathbf{x}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^k \\ \mathbf{x}_1^\top \leftarrow [\mathbf{I}_\ell \otimes \mathbf{D}]^{-1} (\mathbf{t}^\top - \mathbf{W}\mathbf{x}_2^\top, \sigma) \end{array} \right\}$$

Proposition 3.2 (Statistical Closeness of Gaussian Inner Product to Uniform [MR09, MKMS22], *Adapted*). *Let n, m, p be integers such that $m = \mathcal{O}(n \log p)$ and $p > 2$ is prime. Let $\chi > \omega(\sqrt{\log m})$ be a Gaussian width parameter, then for a fixed nonzero large vector $\mathbf{t} \in \mathbb{Z}_p^m$, the statistical distance between the following distributions is $\mathsf{negl}(n)$.*

$$\left\{ \left(\mathbf{t}, \boxed{\langle \mathbf{t}, \mathbf{r} \rangle \bmod p} \right) \ : \ \mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m \right\} \quad \textit{and} \quad \left\{ (\mathbf{t}, \boxed{u}) \ : \ u \xleftarrow{\$} \mathbb{Z}_p \right\}$$

This result is an extension of the Leftover Hash Lemma.

Lemma 3.6 (Homomorphic Encodings [GSW13, BGG+14]). *Let λ be a security parameter and n, m, and p be lattice parameters such that $m \geq n \log p$. Let ℓ be an input length and let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ that can be computed by a boolean circuit of depth at most d. Then, there exists a pair of efficient and deterministic algorithms $(\mathsf{EvalF}, \mathsf{EvalFX})$ with the following properties:*

- $\mathbf{P}_f \leftarrow \mathsf{EvalF}(\mathbf{P}, f)$*: On input a matrix $\mathbf{P} \in \mathbb{Z}_p^{n \times \ell m}$ and a function $f \in \mathcal{F}$, the input-independent evaluation algorithm outputs a matrix $\mathbf{P}_f \in \mathbb{Z}_p^{n \times m}$.*
- $\mathbf{H}_{\mathbf{P},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\mathbf{P}, f, \mathbf{x})$*: On input a matrix $\mathbf{P} \in \mathbb{Z}_p^{n \times \ell m}$, a function $f \in \mathcal{F}$, and an input $\mathbf{x} \in \{0,1\}^\ell$, the input-dependent evaluation algorithm outputs a matrix $\mathbf{H}_{\mathbf{P},f,\mathbf{x}} \in \mathbb{Z}_p^{\ell m \times m}$.*

Moreover, for all security parameters $\lambda \in \mathbb{N}$, matrices $\mathbf{P} \in \mathbb{Z}_p^{n \times \ell m}$, all functions $f \in \mathcal{F}$, and all inputs $\mathbf{x} \in \{0,1\}^\ell$, the matrices $\mathbf{P}_f = \mathsf{EvalF}(\mathbf{P}, f)$ and $\mathbf{H}_{\mathbf{P},f,\mathbf{x}} = \mathsf{EvalFX}(\mathbf{P}, f, \mathbf{x})$ satisfy the following properties:

$$\|\mathbf{H}_{\mathbf{P},f,\mathbf{x}}\| \leq m^{\mathcal{O}(d)},$$

$$(\mathbf{P} - \mathbf{x} \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{P},f,\mathbf{x}} = \mathbf{P}_f - f(\mathbf{x}) \cdot \mathbf{G}.$$

3.2 Matrix Commitment

The matrix commitment scheme, introduced in [Wee25] and further modified in [WW25], allows one to generate a commitment matrix $\mathbf{C} \in \mathbb{Z}_p^{n\times m}$ to a matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times L}$, with respect to a matrix $\mathbf{D} \in \mathbb{Z}_p^{n\times m}$ such that $\mathbf{C}\cdot\mathbf{V} = \mathbf{M} - \mathbf{D}\cdot\mathbf{Z}$ where, $\mathbf{V} \in \mathbb{Z}_p^{m\times L}$ is the *fixed* verification matrix depending only on the dimension L and $\mathbf{Z} \in \mathbb{Z}_p^{m\times L}$ is a short opening.

Let n, m, and p be lattice parameters such that $m \geq 2n\log p$. In the following, we denote public parameters for commitment by $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$, where $\mathbf{D} \in \mathbb{Z}_p^{n\times m}$, $\mathbf{W} \in \mathbb{Z}_p^{2m^2 n\times m}$, $\mathbf{T} \in \mathbb{Z}_p^{(2m^2+1)m\times 2m^3}$, and $[\mathbf{I}_{2m^2}\otimes\mathbf{D} \mid \mathbf{W}]\cdot\mathbf{T} = \mathbf{I}_{2m^2}\otimes\mathbf{G}$.

Lemma 3.7 (Matrix Commitment [Wee25, WW25]). *Let n, m and p be lattice parameters such that $m \geq 2n\log p$, then there exist three efficient and deterministic algorithms $\left(\mathsf{Com}^{\mathsf{mat}}, \mathsf{Ver}^{\mathsf{mat}}, \mathsf{Open}^{\mathsf{mat}}\right)$ with the following syntax:*

- $(\mathbf{C}) \leftarrow \mathsf{Com}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$*: Given the parameters $\mathsf{pp}_{\mathsf{com}}$ and a matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times L}$, the algorithm outputs a matrix $\mathbf{C} \in \mathbb{Z}_p^{n\times m}$.*
- $(\mathbf{V}) \leftarrow \mathsf{Ver}^{\mathsf{mat}}\left(\mathsf{pp}_{\mathsf{com}}, 1^L\right)$*: Given the parameters $\mathsf{pp}_{\mathsf{com}}$ and a length parameter L, the algorithm outputs a matrix $\mathbf{V} \in \mathbb{Z}_p^{m\times L}$.*
- $(\mathbf{Z}) \leftarrow \mathsf{Open}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$*: Given the parameters $\mathsf{pp}_{\mathsf{com}}$ and a matrix $\mathbf{M} \in \mathbb{Z}_p^{n\times L}$, the algorithm outputs a matrix $\mathbf{Z} \in \mathbb{Z}_p^{m\times L}$.*

Moreover, for all length parameters $L \in \mathbb{N}$, all matrices $\mathbf{M} \in \mathbb{Z}_p^{n\times L}$, all public parameters for commitment $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$, where $\mathbf{D} \in \mathbb{Z}_p^{n\times m}$, $\mathbf{W} \in \mathbb{Z}_p^{2m^2 n\times m}$, $\mathbf{T} \in \mathbb{Z}_p^{(2m^2+1)m\times 2m^3}$ and $[\mathbf{I}_{2m^2}\otimes\mathbf{D} \mid \mathbf{W}]\cdot\mathbf{T} = \mathbf{I}_{2m^2}\otimes\mathbf{G}$, and setting $\mathbf{C} = \mathsf{Com}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$, $\mathbf{V} = \mathsf{Ver}^{\mathsf{mat}}\left(\mathsf{pp}_{\mathsf{com}}, 1^L\right)$ and $\mathbf{Z} = \mathsf{Open}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$, we have

$$\mathbf{C}\cdot\mathbf{V} = \mathbf{M} - \mathbf{D}\cdot\mathbf{Z},$$

with $\|\mathbf{V}\| \leq \mathcal{O}\left(\|\mathbf{T}\|\cdot m^4\log p\right)$ and $\|\mathbf{Z}\| \leq \mathcal{O}\left(\|\mathbf{T}\|\cdot m^7\log p\log L\right)$. The running time of all three algorithms is bounded above by $L\cdot\log L\cdot\mathsf{poly}(m)$.

3.3 Publicly-Sampleable Projective PRGs

A Publicly-Sampleable Projective Pseudorandom Generator [ABI+23, HWW25] is a tuple of four efficient algorithms $\prod_{\mathsf{pPRG}} = (\mathsf{pSetup}, \mathsf{pSamp}, \mathsf{pProject}, \mathsf{pEval})$ with the following syntax:

- $(\mathsf{pp}, \delta) \leftarrow \mathsf{pSetup}\left(1^\lambda, 1^l\right)$: Taking as input the security parameter λ and the output length $l \in \mathbb{N}$, the setup algorithm returns the public parameters pp and a seed δ.

- $(\delta_S) \leftarrow \mathsf{pSamp}(\mathsf{pp}, S)$**:** Given the public parameters pp and a subset $S \subseteq [l]$ as input, this public sampling algorithm produces a simulated projected seed δ_S.
- $(\delta_S) \leftarrow \mathsf{pProject}(\mathsf{pp}, \delta, S)$**:** On input of public parameters pp, a seed δ and a subset $S \subseteq [l]$, the projection algorithm outputs the projected seed δ_S.
- $(y_j) \leftarrow \mathsf{pEval}(\mathsf{pp}, \delta_S, S, j)$**:** Given inputs pp, a projected seed δ_S, a subset $S \subseteq [l]$, and an index $j \in S$, the evaluation algorithm deterministically returns a bit $y_j \in \{0,1\}$.

Such a construction is expected to satisfy the following properties:

• **Correctness.** For all $\lambda, l \in \mathbb{N}$, subsets $S \subseteq [l]$ with $j \in S$, the following holds:

$$\Pr\left[\mathsf{pEval}\left(\mathsf{pp}, \delta_{[l]}, [l], j\right) = \mathsf{pEval}\left(\mathsf{pp}, \delta_S, S, j\right) \middle| \begin{array}{l} (\mathsf{pp}, \delta) \leftarrow \mathsf{pSetup}(1^\lambda, 1^l), \\ \delta_{[l]} \leftarrow \mathsf{pProject}(\mathsf{pp}, \delta, [l]), \\ \delta_S \leftarrow \mathsf{pProject}(\mathsf{pp}, \delta, S) \end{array}\right] = 1$$

• **Succinctness.** For any $\lambda, l \in \mathbb{N}$, any subset $S \subseteq [l]$, and any δ_S returned by the pSamp algorithm, the size of δ_S is bounded by $\mathsf{poly}(\lambda, \log l)$.

• **Sampling Indistinguishability.** For the security parameter λ, a bit $b \in \{0,1\}$, and an efficient adversary $\mathcal{A}$, consider the experiment $\mathsf{Exp}^{(b)}_{samp}(1^\lambda, \mathcal{A})$ as follows:

- **Setup Phase:** The adversary is given 1^λ and the outputs $l \in \mathbb{N}$. The challenger runs $(\mathsf{pp}, \delta) \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ and sends pp to the adversary.
- **Challenge Phase:** $\mathcal{A}$ sends the challenge set $S \subseteq [l]$. The projected key δ_S is constructed as follows:
  (i) If $b = 0$, the challenger computes $\delta_S \leftarrow \mathsf{pSamp}(\mathsf{pp}, S)$.
  (ii) If $b = 1$, the challenger computes $\delta_S \leftarrow \mathsf{pProject}(\mathsf{pp}, \delta, S)$.
  The challenger sends δ_S to $\mathcal{A}$.
- **Guess Phase:** The adversary outputs a guess $b' \in \{0,1\}$.

The adversary's advantage is defined as:

$$\left|\Pr\left[\mathsf{Exp}^{(0)}_{samp}(1^\lambda, \mathcal{A}) = 1\right] - \Pr\left[\mathsf{Exp}^{(1)}_{samp}(1^\lambda, \mathcal{A}) = 1\right]\right|$$

We say $\prod_{\mathsf{pPRG}}$ satisfies *sampling indistinguishability* if the above advantage is negligible in λ for all efficient adversaries $\mathcal{A}$. If the same bound holds even for unbounded adversaries, the notion is said to be *statistically indistinguishable*. If the difference is exactly zero, it achieves *perfect sampling indistinguishability*.

Lemma 3.8 (Publicly-Sampleable Projective PRG from LWE [HWW25]). *There exists a Publicly-Sampleable Projective PRG* $\prod_{\mathsf{pPRG}} = (\mathsf{pSetup}, \mathsf{pSamp}, \mathsf{pProject}, \mathsf{pEval})$ *with linear-size public parameters that satisfy correctness, succinctness, and sampling indistinguishability, under the Learning With Errors (LWE) assumption.*

3.4 Distributed Broadcast Encryption

We define a Distributed Broadcast Encryption (DBE) [BW06, KMW23, CW24, WW25] scheme.

Syntax. Let λ be the security parameter and N be the total number of system users. An N-user DBE is a tuple of four efficient algorithms, $\prod_{\mathsf{DBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

- $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$. This algorithm takes the security parameter λ and the total number of users N (in binary) as inputs and outputs public parameters PP.
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, i\right)$. The key-generation algorithm takes the public parameters PP and an index $i \in [N]$ as inputs to output a public-secret key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$.
- $(\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \{(i, \mathsf{pk}_i)\}_{i\in\mathcal{S}}, \mu\right)$. The broadcaster runs the encryption algorithm by taking as inputs the public parameters PP, a collection of public keys $\{\mathsf{pk}_i\}_{i\in\mathcal{S}}$, where $\mathcal{S} \subseteq [N]$ is the authorized receivers' set and a message $\mu \in \{0,1\}$ to output a ciphertext CT.
- $(\mu) \leftarrow \mathsf{Dec}\left(\mathsf{PP}, \mathcal{S}, \{(i, \mathsf{pk}_i)\}_{i\in\mathcal{S}}, \mathsf{CT}, (j, \mathsf{sk}_j)\right)$. On input the public parameters PP, a collection of public keys $\{\mathsf{pk}_i\}_{i\in\mathcal{S}}$, a ciphertext CT, a secret key sk_j for an index $j \in \mathcal{S}$, the decryption algorithm outputs the message $\mu \in \{0,1\}$.

We require that the tuple $\prod_{\mathsf{DBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies the following properties:

- **Correctness.** For all $\lambda, N \in \mathbb{N}$, all subsets $\mathcal{S} \subseteq [N]$, all indices $i \in \mathcal{S}$, all public parameters PP in support of $\mathsf{Setup}(1^\lambda, 1^N)$, all key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ in support of $\mathsf{KeyGen}(\mathsf{PP}, i)$, all $\mu \in \{0,1\}$, it holds that

  $$\Pr\left[\mu \leftarrow \mathsf{Dec}\left(\mathsf{PP}, \{(i, \mathsf{pk}_i)\}_{i\in\mathcal{S}}, \mathsf{Enc}\left(\mathsf{PP}, \{(i, \mathsf{pk}_i)\}_{i\in\mathcal{S}}, \mu\right), (j, \mathsf{sk}_j)\right)\right] = 1 - \mathsf{negl}(\lambda).$$

- **Semi-Static Security.** For all security parameters λ, the total number of users N and a bit $b \in \{0,1\}$, we define the semi-static security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:
  - **Initialization:** Given the security parameter λ and the total number of users N, the adversary $\mathcal{A}$ sends a set $\mathcal{S}^* \subseteq [N]$ to the challenger $\mathcal{C}$.
  - **Setup & Key-Generation:** The challenger $\mathcal{C}$ generates the public parameters $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$, and then for all $i \in \mathcal{S}^*$, it samples a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{PP}, i)$ and sends $(\mathsf{PP}, \{\mathsf{pk}_i\}_{i\in\mathcal{S}^*})$ to $\mathcal{A}$.
  - **Challenge:** Given a challenge set $\mathcal{S} \subseteq \mathcal{S}^*$ by $\mathcal{A}$, the challenger $\mathcal{C}$ computes the challenge ciphertext $(\mathsf{CT}_b) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \{(i, \mathsf{pk}_i)\}_{i\in\mathcal{S}}, b\right)$ and sends CT_b to $\mathcal{A}$.

– **Output:** At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

We say that a DBE scheme is semi-statically secure for all $\lambda \in \mathbb{N}$ if for all bounds $N \leq 2^\lambda$, and all efficient adversaries $\mathcal{A}$, we have

$$\left| \Pr\left[b' = b\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

- **Adaptive Security.** For all security parameters λ, the total number of users N, and a bit $b \in \{0,1\}$, we define the adaptive security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows:

  – **Setup:** Given the security parameter λ and the total number of users N, the challenger $\mathcal{C}$ generates the public parameters $\mathsf{PP} \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$, and sends PP to the adversary $\mathcal{A}$. It also initializes the following:

  (a) an empty dictionary $\mathtt{D}$ to store key information,

  (b) a counter $\mathsf{ctr} = 0$ to track key-generation queries,

  (c) an empty set $\mathtt{C}$ to keep track of corrupted keys.

  – **Queries:** The adversary $\mathcal{A}$ can adaptively make the following queries:

  (a) $\mathsf{Key\text{-}generation\ query}$: Taking as input a index $i \in [N]$, the challenger generates a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{PP}, i)$, sends pk_i to $\mathcal{A}$, registers $(i, \mathsf{pk}_i, \mathsf{sk}_i)$ at $\mathtt{D}[\mathsf{ctr}]$, and increments ctr.

  (b) $\mathsf{Corruption\ query}$: Given an index $i \in [\mathsf{ctr}]$, the challenger reveals $\mathtt{D}[i]$ and adds i to the set $\mathtt{C}$.

  – **Challenge:** In the challenge phase, the adversary $\mathcal{A}$ selects a challenge set $\mathcal{S} \subseteq [\mathsf{ctr}]$. The challenger verifies the following:

  (a) $\mathcal{S} \cap \mathtt{C} = \emptyset$ to ensure that no corrupted users are in the challenge set,

  (b) For each $j \in \mathcal{S}$, let $\mathtt{D}[j] = \left(i_j, \mathsf{pk}_j, \mathsf{sk}_j\right)$. Then, the indices i_j must all be distinct to make sure that at most one public key is associated with each slot.

  If any check fails, the challenger aborts the game. Otherwise, it computes the challenge ciphertext $\mathsf{CT}_b \leftarrow \mathsf{Encrypt}(\mathsf{PP}, \mathcal{S}, \{(i_j, \mathsf{pk}_j)\}_{j \in \mathcal{S}}, b)$, and sends CT_b to $\mathcal{A}$.

  – **Output:** At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

  We say that a DBE scheme is adaptively-secure for all $\lambda \in \mathbb{N}$ if for all bounds $N \leq 2^\lambda$, and all efficient adversaries $\mathcal{A}$, we have

$$\left| \Pr\left[b' = b\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

- **Succinctness.** For all $\lambda, N \in \mathbb{N}$, all subsets $\mathcal{S} \subseteq [N]$, all indices $i \in [N]$, all public parameters PP in support of $\mathsf{Setup}(1^\lambda, N)$, all key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ in support of $\mathsf{KeyGen}(\mathsf{PP}, i)$, all $\mu \in \{0,1\}$, and all ciphertexts CT in the support of $\mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{(j, \mathsf{pk}_j)\right\}_{j \in \mathcal{S}}, \mu\right)$, it holds that $|\mathsf{CT}| \leq \mathsf{poly}(\lambda, \log N)$.

Encrypting Long Messages. Similar to the works of [CW24,CHW25,WW25], we are also considering a simple setting, where the ciphertext encrypts a single bit $\mu \in \{0,1\}$. To encrypt longer messages, we can use a *hybrid* system using symmetric-key encryption schemes. Namely, to encrypt a message $\mathrm{M} \in \{0,1\}^l$, the encryption algorithm samples a symmetric-key $k \in \{0,1\}^{\mathsf{poly}(\lambda)}$ and encrypts the bits of k using the DBE, and then encrypts the message M using the symmetric-key k.

4 Partitionable Matrix Commitment

The matrix commitment scheme, introduced in [Wee25] and extended in [WW25], enables one to commit to a matrix $\mathbf{M} \in \mathbb{Z}_p^{n \times L}$ with respect to another matrix $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$, resulting in a commitment matrix $\mathbf{C} \in \mathbb{Z}_p^{n \times m}$ that satisfies the following relation:

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z},$$

where $\mathbf{V} \in \mathbb{Z}_p^{m \times L}$ is a *fixed* verification matrix and $\mathbf{Z} \in \mathbb{Z}_p^{m \times L}$ is a short opening.

Although powerful, the above construction does not directly suit the needs of our DBE scheme. Specifically, our application requires committing to structured matrices $\mathbf{M} \in \mathbb{Z}_p^{n \times L}$, where $L = r \cdot s$ and

$$\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2 \mid \cdots \mid \mathbf{M}_s],$$

with each block $\mathbf{M}_t \in \mathbb{Z}_p^{n \times r}$ for $t \in [s]$. To facilitate efficient partition-wise verification and opening, we seek a block-partitionable interface to the original commitment scheme. That is, we require the verification matrix $\mathbf{V}$ and the opening matrix $\mathbf{Z}$ to be partitioned as

$$\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_s], \quad \mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_s],$$

where $\mathbf{V}_t, \mathbf{Z}_t \in \mathbb{Z}_p^{m \times r}$ for each $t \in [s]$, so that the following relation holds:

$$\mathbf{C} \cdot \mathbf{V}_t = \mathbf{M}_t - \mathbf{D} \cdot \mathbf{Z}_t, \quad \forall t \in [s].$$

We term this structure-preserving enhancement as Partitionable Matrix Commitment (pmat). Crucially, this interface does not introduce changes to the

core commitment algorithm, which remains identical to the construction of [Wee25, WW25]. Our contribution lies in enabling modular access to the committed blocks via decomposed verification and opening matrices. This enhancement supports partial openings, efficient reuse, and compatibility with block-structured data. All algorithmic operations in partitionable matrix commitment inherit the efficiency bounds of the original construction and can be computed in time bounded by $L \cdot \log L \cdot \mathsf{poly}(m)$. We formalize the interface and guarantees in the Definition 4.1.

Let the public parameters for matrix commitment be $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$, where $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$, $\mathbf{W} \in \mathbb{Z}_p^{2m^2 n \times m}$, $\mathbf{T} \in \mathbb{Z}_p^{(2m^2+1)m \times 2m^3}$, and $[\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}] \cdot \mathbf{T} = \mathbf{I}_{2m^2} \otimes \mathbf{G}$.

Definition 4.1 (Partitionable Matrix Commitment). *Let n, m, and p be lattice parameters such that $m \geq 2n \log p$. Then there exist three efficient and deterministic algorithms $\left(\mathsf{Com}^{\mathsf{pmat}}, \mathsf{Ver}^{\mathsf{pmat}}, \mathsf{Open}^{\mathsf{pmat}}\right)$ with the following syntax:*

- $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$*: Given $\mathbf{M} = [\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_s] \in \mathbb{Z}_p^{n \times r \cdot s}$, compute and output $\mathbf{C} \leftarrow \mathit{Com}^{\mathit{mat}}(\mathit{pp}_{\mathit{com}}, \mathbf{M})$.*
- $\mathbf{V}_t \leftarrow \mathsf{Ver}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, 1^{r \cdot s}, t)$*: Compute $\mathbf{V} \leftarrow \mathit{Ver}^{\mathit{mat}}(\mathit{pp}_{\mathit{com}}, 1^{r \cdot s})$ and return the t^{th} block $\mathbf{V}_t \in \mathbb{Z}_p^{m \times r}$.*
- $\mathbf{Z}_t \leftarrow \mathsf{Open}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}, t)$*: Compute $\mathbf{Z} \leftarrow \mathit{Open}^{\mathit{mat}}(\mathit{pp}_{\mathit{com}}, \mathbf{M})$ and return the t^{th} block $\mathbf{Z}_t \in \mathbb{Z}_p^{m \times r}$.*

Moreover, for all $\mathbf{M} \in \mathbb{Z}_p^{n \times r \cdot s}$, all $t \in [s]$, and all valid public parameters $\mathsf{pp}_{\mathsf{com}}$, we have

$$\mathbf{C} \cdot \mathbf{V}_t = \mathbf{M}_t - \mathbf{D} \cdot \mathbf{Z}_t.$$

The norm bounds remain identical to those in the original scheme:

$$\|\mathbf{V}_t\| \leq \mathcal{O}(\|\mathbf{T}\| \cdot m^4 \log p), \quad \|\mathbf{Z}_t\| \leq \mathcal{O}(\|\mathbf{T}\| \cdot m^7 \log p \log L),$$

and the running time of all algorithms is bounded above by $L \cdot \log L \cdot \mathit{poly}(m)$.

Lemma 4.1 (Correctness of Partitionable Matrix Commitment). *Let the algorithms $(\mathsf{Com}^{\mathsf{pmat}}, \mathsf{Ver}^{\mathsf{pmat}}, \mathsf{Open}^{\mathsf{pmat}})$ be as defined in the Partitionable Matrix Commitment Definition 4.1, which are constructed from a correct matrix commitment algorithms $(\mathsf{Com}^{\mathsf{mat}}, \mathsf{Ver}^{\mathsf{mat}}, \mathsf{Open}^{\mathsf{mat}})$. Let $\mathsf{pp}_{\mathsf{com}}$ be valid public parameters and let $\mathbf{M} = [\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_s]$, where $\mathbf{M}_t \in \mathbb{Z}_p^{n \times r}$ for each $t \in [s]$. The correctness of the underlying matrix commitment scheme guarantees that for a commitment $\mathbf{C} = \mathsf{Com}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$, the relation $\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}$ holds, where $\mathbf{V} = \mathsf{Ver}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, 1^{r \cdot s})$ and $\mathbf{Z} = \mathsf{Open}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$. This implies the correctness of the partitionable matrix commitment for each block $t \in [s]$, such that:*

$$\mathbf{C} \cdot \mathbf{V}_t = \mathbf{M}_t - \mathbf{D} \cdot \mathbf{Z}_t,$$

where $\mathbf{V}_t$ *and* $\mathbf{Z}_t$ *are the respective* t^{th} *blocks of* $\mathbf{V}$ *and* $\mathbf{Z}$ *as returned by* $\mathsf{Ver}^{\mathsf{pmat}}$ *and* $\mathsf{Open}^{\mathsf{pmat}}$.

Proof. We will prove the Lemma 4.1 by induction on the number of blocks, s.

– **Base Case** ($s = 1$): For $s = 1$, the matrix $\mathbf{M}$ consists of a single block, $\mathbf{M} = [\mathbf{M}_1]$. The partitionable matrix commitment algorithms are defined to be equivalent to the original matrix commitment algorithms:

- $\mathbf{C} = \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}_1) = \mathsf{Com}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}_1)$
- $\mathbf{V}_1 = \mathsf{Ver}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, 1^r, 1) = \mathsf{Ver}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, 1^r)$
- $\mathbf{Z}_1 = \mathsf{Open}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}_1, 1) = \mathsf{Open}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}_1)$

The statement we need to prove is $\mathbf{C} \cdot \mathbf{V}_1 = \mathbf{M}_1 - \mathbf{D} \cdot \mathbf{Z}_1$. This is precisely the correctness guarantee of the underlying matrix commitment scheme. Thus, the base case holds.

– **Inductive Hypothesis**: Assume that the Lemma 4.1 holds for any matrix composed of $s = k$ blocks for some $k \geq 1$. That is, for any matrix $\mathbf{M} = [\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_k]$, given a commitment $\mathbf{C}$ and corresponding verification and opening matrices $\mathbf{V}$ and $\mathbf{Z}$ such that $\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}$, it implies that $\mathbf{C} \cdot \mathbf{V}_t = \mathbf{M}_t - \mathbf{D} \cdot \mathbf{Z}_t$ for all $t \in [k]$, where $\mathbf{V}_t$ and $\mathbf{Z}_t$ are the respective t^{th} blocks of $\mathbf{V}$ and $\mathbf{Z}$.

– **Inductive Step** ($s = k + 1$): Let $\mathbf{M} = [\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_k \mid \mathbf{M}_{k+1}]$ be a matrix with $k+1$ blocks. Let $\mathbf{C} = \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$. By definition, this is equivalent to $\mathbf{C} = \mathsf{Com}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$.

From the underlying matrix commitment, we know that $\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}$, where $\mathbf{V} = \mathsf{Ver}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, 1^{r(k+1)})$ and $\mathbf{Z} = \mathsf{Open}^{\mathsf{mat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$. We can partition $\mathbf{M}$, $\mathbf{V}$, and $\mathbf{Z}$ into two parts: a k-block part and a final 1-block part.

- $\mathbf{M} = [\mathbf{M}' \mid \mathbf{M}_{k+1}]$, where $\mathbf{M}' = [\mathbf{M}_1 \mid \cdots \mid \mathbf{M}_k]$.
- $\mathbf{V} = [\mathbf{V}' \mid \mathbf{V}_{k+1}]$, where $\mathbf{V}'$ contains the first k blocks of $\mathbf{V}$.
- $\mathbf{Z} = [\mathbf{Z}' \mid \mathbf{Z}_{k+1}]$, where $\mathbf{Z}'$ contains the first k blocks of $\mathbf{Z}$.

Substituting these into the equation:

$$\mathbf{C} \cdot [\mathbf{V}' \mid \mathbf{V}_{k+1}] = [\mathbf{M}' \mid \mathbf{M}_{k+1}] - \mathbf{D} \cdot [\mathbf{Z}' \mid \mathbf{Z}_{k+1}]$$

Using the properties of block matrix multiplication, this expands to:

$$[\mathbf{C} \cdot \mathbf{V}' \mid \mathbf{C} \cdot \mathbf{V}_{k+1}] = [\mathbf{M}' - \mathbf{D} \cdot \mathbf{Z}' \mid \mathbf{M}_{k+1} - \mathbf{D} \cdot \mathbf{Z}_{k+1}]$$

For this equality to hold, the corresponding blocks must be equal. This gives us two equations:

1. $\mathbf{C} \cdot \mathbf{V}_{k+1} = \mathbf{M}_{k+1} - \mathbf{D} \cdot \mathbf{Z}_{k+1}$
2. $\mathbf{C} \cdot \mathbf{V}' = \mathbf{M}' - \mathbf{D} \cdot \mathbf{Z}'$

The first equation proves the lemma for the block $t = k+1$. The second equation matches our inductive hypothesis for the k-block matrix $\mathbf{M}'$. Therefore, by the inductive hypothesis, we can conclude that $\mathbf{C} \cdot \mathbf{V}_t = \mathbf{M}_t - \mathbf{D} \cdot \mathbf{Z}_t$ for all $t \in [k]$.

Since the property holds for $t \in [k]$ and for $t = k+1$, it holds for all $t \in [k+1]$. By the principle of mathematical induction, the lemma is valid for all $s \geq 1$.

Norm Bounds. Let $L = r \cdot s$, then as observed from the structure of the algorithms $\mathsf{Ver}^{\mathsf{pmat}}$ and $\mathsf{Open}^{\mathsf{pmat}}$, it is easy to see that

$$\|\mathbf{V}_t\| \leq \|\mathbf{V}\| \leq \mathcal{O}(\|\mathbf{T}\| \cdot m^4 \log p) \quad \text{and} \quad \|\mathbf{Z}_t\| \leq \|\mathbf{Z}\| \leq \mathcal{O}(\|\mathbf{T}\| \cdot m^7 \log p \log L).$$

Running Time. Let $L = r \cdot s$, then the running times of the three algorithms in the partitionable matrix commitment scheme are as follows:

- $\mathsf{Com}^{\mathsf{pmat}}$: According to Definition 4.1, its running time coincides with that of $\mathsf{Com}^{\mathsf{mat}}$, which is bounded by $L \cdot \log L \cdot \mathsf{poly}(m)$.
- $\mathsf{Ver}^{\mathsf{pmat}}$: As specified in Definition 4.1, its running time corresponds to $\mathsf{Ver}^{\mathsf{mat}}$ together with the additional cost of identifying the t^{th} block. The total time is therefore at most $\log s + L \cdot \log L \cdot \mathsf{poly}(m)$.
- $\mathsf{Open}^{\mathsf{pmat}}$: Analogously, this procedure requires the running time of $\mathsf{Open}^{\mathsf{mat}}$ alongwith the block-location step, yielding the same upper bound of $\log s + L \cdot \log L \cdot \mathsf{poly}(m)$.

Remark 4.1 (Generalization to Arbitrary Column Subsets). *The partitionable matrix commitment demonstrated in Definition 4.1 for contiguous partitions is a special case of a more general principle. The correctness equation* $\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z}$ *holds column-by-column. Consequently, the commitment is not just modular with respect to blocks, but is verifiable for any arbitrary subset of columns.*

More formally, let $\mathcal{I} \subseteq [L]$ *be any set of column indices. Let* $\mathbf{M}_{\mathcal{I}}$, $\mathbf{V}_{\mathcal{I}}$, *and* $\mathbf{Z}_{\mathcal{I}}$ *denote the sub-matrices formed by selecting only the columns corresponding to the indices in* $\mathcal{I}$. *Then the following relation holds:*

$$\mathbf{C} \cdot \mathbf{V}_{\mathcal{I}} = \mathbf{M}_{\mathcal{I}} - \mathbf{D} \cdot \mathbf{Z}_{\mathcal{I}}$$

5 Semi-Statically Secure DBE in Plain Model

In this section, we give our construction of a semi-statically secure distributed broadcast encryption scheme from ℓ-succinct LWE in the plain model, where $\ell = \mathsf{poly}(\lambda, \log N)$, λ is a security parameter and N is a bound on the number of users. For each integer $i \in [N]$, we associate with it a unique canonical vector $\boldsymbol{\gamma}_i \in \{0,1\}^q$, where $q = \lceil \log N \rceil + 1$. For example, we can take $\boldsymbol{\gamma}_i$ as the binary representation of i in q bits.

5.1 Construction of Unbounded Optimal DBE from Succinct LWE

Here we consider $N \leq 2^\lambda$, and propose the construction of our DBE scheme that achieves optimal parameter sizes and is proven semi-statically secure in the plain model in subsequent sections.

Construction 1. Let λ be a security parameter, N be a bound on the number of users, and (n, m, p, χ') be LWE parameters (possibly, functions of λ and N). Let χ, σ be Gaussian width parameters, then we construct our DBE scheme, $\prod_{\mathsf{DBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, as follows:

- $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$: Taking the security parameter λ and the bound N on the total number of users in the system as inputs, the setup algorithm executes the following steps:

  1. Sets $q = \lceil \log N \rceil + 1$ and samples $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n \times q}$ and $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$.
  2. Samples the pair $(\mathbf{D}, \mathbf{T_D}) \leftarrow \mathsf{TrapGen}\left(1^n, 1^m, p\right)$, and chooses $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{2m^2 n \times m}$.
  3. Runs $\mathbf{T} \leftarrow \mathsf{SamplePre}\left(\left[\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}\right], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathbf{T_D} \\ 0 \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma\right)$ and sets $\mathsf{pp_{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$.

  The algorithm outputs the public parameters $\mathsf{PP} = (N, \mathsf{pp_{com}}, \mathbf{B}, \mathbf{L})$.

- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, i\right)$: Given the public parameters PP and an index $i \in [N]$, the key-generation algorithm proceeds as follows:

  1. Samples $\mathbf{h}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m$ and sets $\mathbf{d}_i^\top = \mathbf{L} \cdot \mathbf{h}_i^\top + \mathbf{B} \cdot \boldsymbol{\gamma}_i^\top \in \mathbb{Z}_p^n$.

  The algorithm finally sets the public key $\mathsf{pk}_i = \mathbf{d}_i$ and the secret key $\mathsf{sk}_i = \mathbf{h}_i$.

- $(\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k \in [\varsigma]}, \mu\right)$: On input the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ with corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k \in [\varsigma]}$, and a one-bit message μ, the broadcaster runs the encryption algorithm as follows:

1. Samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^m$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$, and $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$.
2. Sets $\mathbf{g}' = [\mathbf{g} \mid 0^m]$, where $\mathbf{g} = [1, 2, \cdots, 2^{\lceil \log p \rceil - 1}, 0, 0, \cdots, 0] \in \mathbb{Z}^m$ is a gadget vector.
3. For all $k \in [\varsigma]$, sets $\mathbf{M}_k = \left[\mathbf{d}_{i_k}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_k}^\top\right]$, and forms the matrix $\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2 \mid \cdots \mid \mathbf{M}_\varsigma] \in \mathbb{Z}_p^{n\times\varsigma}$.
4. Computes the commitment matrix $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$.

The final ciphertext is as follows:

$$\mathsf{CT} = \left(\overbrace{\mathbf{sD} + \mathbf{e}}^{\mathbf{c}_0},\ \overbrace{\mathbf{s}(\mathbf{L} \mid \mathbf{A}) + \mu\mathbf{g}' + \mathbf{eK}_1}^{\mathbf{c}_1},\ \overbrace{\mathbf{s}(\mathbf{C} + \mathbf{A}) + \mathbf{eK}_2}^{\mathbf{c}_2} \right).$$

- $(\mu) \leftarrow \mathsf{Dec}\left(\mathsf{PP}, \mathcal{S}, \left\{(i_k, \mathsf{pk}_{i_k})\right\}_{k\in[\varsigma]}, \mathsf{CT}, (j, \mathsf{sk}_j)\right)$: Taking as input the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\}$, corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$, a valid ciphertext CT, and the secret key sk_j of the user indexed by j such that $j \in \mathcal{S}$, *i.e.*, $j = i_k$ for some $k \in [\varsigma]$, the user j runs the decryption algorithm as follows:
  1. Finds $k \in [\varsigma]$ such that $j = i_k$ and reconstructs $\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2 \mid \cdots \mid \mathbf{M}_\varsigma]$.
  2. Samples $\mathbf{V}_k \leftarrow \mathsf{Ver}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, 1^\varsigma, k)$ and $\mathbf{Z}_k \leftarrow \mathsf{Open}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}, k)$.

  The decryptor finally outputs

$$\mu' = \mathsf{round}_{\beta_0}\left(\left|\mathbf{c}_1 \begin{pmatrix} \mathbf{h}_{i_k}^\top \\ \mathbf{V}_k \end{pmatrix} - (\mathbf{c}_2\mathbf{V}_k + \mathbf{c}_0\mathbf{Z}_k)\right|\right),$$

  where β_0 is the error bound and $\mathsf{round}_{\beta_0}(x) = \begin{cases} 0 & \text{if } |x| \le \beta_0 \\ 1 & \text{otherwise.} \end{cases}$

5.2 Proof of Correctness and Succinctness

In this section, we provide the correctness and succinctness of the proposed Construction 1 in the form of provable theorems.

Theorem 5.1 (Correctness). *Suppose $\lambda \in \mathbb{N}$ is the security parameter, $N \in \mathbb{N}$ is the total number of users, $n \ge \lambda$, the modulus p is prime, $m = \mathcal{O}(n \log p)$, $\chi \ge m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}\left(m^3 \cdot \log m\right)$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi\chi'\sigma \log p \log N)$, and $p \ge \beta_0 \cdot \lambda^{\omega(1)}$. Assuming that there exists a user indexed by j such that $j \in \mathcal{S}$, then the Construction 1 satisfies the protocol correctness.*

Proof. Let the system public parameters $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}), \mathbf{B}, \mathbf{L})$ be generated from the $\mathsf{Setup}\left(1^\lambda, N\right)$ algorithm. Also, for any index $i \in [N]$, consider $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, i\right)$. Take any set $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ and the corresponding collection of public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$. Now choose a message $\mu \in \{0,1\}$ and let $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2) = (\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mu\right)$. Let $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi'}$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$, $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$, and $\mathbf{g}' = [\mathbf{g} \mid 0^m]$ be sampled during encryption. Let for all $k \in [\varsigma]$, $\mathbf{M}_k = \left[\mathbf{d}_{i_k}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_k}^\top\right]$ and $\mathbf{M} = [\mathbf{M}_1 \mid \mathbf{M}_2 \mid \cdots \mid \mathbf{M}_\varsigma] \in \mathbb{Z}_p^{n\times\varsigma}$ and the commitment matrix $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}\right)$. Now, $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\}$ and since $j \in \mathcal{S}$, we have that $j = i_k$ for some $k \in [\varsigma]$. Then the user with index j samples $\mathbf{V}_k \leftarrow \mathsf{Ver}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, 1^\varsigma, k\right)$ and $\mathbf{Z}_k \leftarrow \mathsf{Open}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}, k\right)$ and computes

$$\begin{aligned}
\mathbf{c}_2\mathbf{V}_k &= \left(\mathbf{s}\left(\mathbf{C}+\mathbf{A}\right)+\mathbf{e}\mathbf{K}_2\right)\mathbf{V}_k \\
&= \mathbf{s}\mathbf{C}\mathbf{V}_k + \mathbf{s}\mathbf{A}\mathbf{V}_k + \underbrace{\mathbf{e}\mathbf{K}_2\mathbf{V}_k}_{\widetilde{\mathbf{e}_2}} \\
\mathbf{c}_0\mathbf{Z}_k &= \left(\mathbf{s}\mathbf{D}+\mathbf{e}\right)\mathbf{Z}_k \\
&= \mathbf{s}\mathbf{D}\mathbf{Z}_k + \underbrace{\mathbf{e}\mathbf{Z}_k}_{\widetilde{\mathbf{e}_0}}
\end{aligned}$$

Now, assume that $\mathbf{K}_1 = \left[\overline{\mathbf{K}_1} \mid \underline{\mathbf{K}_1}\right]$, where $\overline{\mathbf{K}_1}$ and $\underline{\mathbf{K}_1} \in \{0,1\}^{m\times m}$

$$\begin{aligned}
\mathbf{c}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} &= \left(\mathbf{s}(\mathbf{L} \mid \mathbf{A}) + \mu\mathbf{g}' + \mathbf{e}\mathbf{K}_1\right)\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} \\
&= \mathbf{s}(\mathbf{L} \mid \mathbf{A})\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} + \mu\mathbf{g}'\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} + \mathbf{e}\mathbf{K}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} \\
&= \mathbf{s}(\mathbf{L} \mid \mathbf{A})\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} + \mu[\mathbf{g} \mid 0^m]\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} + \mathbf{e}\left[\overline{\mathbf{K}_1} \mid \underline{\mathbf{K}_1}\right]\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} \\
&= \mathbf{s}(\mathbf{L}\mathbf{h}_{i_k}^\top + \mathbf{A}\mathbf{V}_k) + \mu\mathbf{g}\mathbf{h}_{i_k}^\top + \mathbf{e}\left(\overline{\mathbf{K}_1}\mathbf{h}_{i_k}^\top + \underline{\mathbf{K}_1}\mathbf{V}_k\right) \\
&= \mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top + \mathbf{s}\mathbf{A}\mathbf{V}_k + \mu\mathbf{g}\mathbf{h}_{i_k}^\top + \underbrace{\mathbf{e}\overline{\mathbf{K}_1}\mathbf{h}_{i_k}^\top}_{\widetilde{\mathbf{e}_1}} + \underbrace{\mathbf{e}\underline{\mathbf{K}_1}\mathbf{V}_k}_{\widetilde{\mathbf{e}_1'}}
\end{aligned}$$

From the above calculations, we have

$$\begin{aligned}
&\mathbf{c}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} - \left(\mathbf{c}_2\mathbf{V}_k + \mathbf{c}_0\mathbf{Z}_k\right) \\
&\quad= \mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top + \mathbf{s}\mathbf{A}\mathbf{V}_k + \mu\mathbf{g}\mathbf{h}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}_1'} \\
&\qquad\qquad - \left(\mathbf{s}\mathbf{D}\mathbf{Z}_k + \widetilde{\mathbf{e}_0} + \mathbf{s}\mathbf{C}\mathbf{V}_k + \mathbf{s}\mathbf{A}\mathbf{V}_k + \widetilde{\mathbf{e}_2}\right) \\
&\quad= \mathbf{s}\mathbf{L}\mathbf{h}_{i_k}^\top + \mathbf{s}\mathbf{A}\mathbf{V}_k + \mu\mathbf{g}\mathbf{h}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}_1'}
\end{aligned}$$

$$
\begin{aligned}
&\quad - \left(\mathbf{s}\left(\mathbf{DZ}_k + \mathbf{CV}_k\right) + \mathbf{sAV}_k + \widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2}\right) \\
&= \mathbf{sLh}_{i_k}^\top + \mathbf{sAV}_k + \mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} \\
&\quad - \left(\mathbf{sM}_k + \mathbf{sAV}_k + \widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2}\right) \\
&= \mathbf{sLh}_{i_k}^\top + \mathbf{sAV}_k + \mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} \\
&\quad - \left(\mathbf{s}\left[\mathbf{d}_{i_k}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_k}^\top\right] + \mathbf{sAV}_k + \widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2}\right) \\
&= \mathbf{sLh}_{i_k}^\top + \mathbf{sAV}_k + \mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} \\
&\quad - \left(\mathbf{sLh}_{i_k}^\top + \mathbf{sAV}_k + \widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2}\right) \\
&= \cancel{\mathbf{sLh}_{i_k}^\top} + \cancel{\mathbf{sAV}_k} + \mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} \\
&\quad - \cancel{\mathbf{sLh}_{i_k}^\top} - \cancel{\mathbf{sAV}_k} - \widetilde{\mathbf{e}_0} - \widetilde{\mathbf{e}_2} \\
&= \mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} - \widetilde{\mathbf{e}_0} - \widetilde{\mathbf{e}_2}
\end{aligned}
$$

For now, let us assume that $\left|\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right| \leq \beta_0$, then the following holds:

$$
\mathsf{round}_{\beta_0}\left(\left|\mathbf{c}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} - \left(\mathbf{c}_2\mathbf{V}_k + \mathbf{c}_0\mathbf{Z}_k\right)\right|\right) = \mathsf{round}_{\beta_0}\left(\left|\mu\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} - \widetilde{\mathbf{e}_0} - \widetilde{\mathbf{e}_2}\right|\right).
$$

– **Case - I:** If $\mu = 0$, then we have

$$
\begin{aligned}
\mathsf{round}_{\beta_0}\left(\left|\mathbf{c}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} - \left(\mathbf{c}_2\mathbf{V}_k + \mathbf{c}_0\mathbf{Z}_k\right)\right|\right) &= \mathsf{round}_{\beta_0}\left(\left|\widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} - \widetilde{\mathbf{e}_0} - \widetilde{\mathbf{e}_2}\right|\right) \\
&= \mathsf{round}_{\beta_0}\left(\left|\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right|\right) \\
&= 0.
\end{aligned}
$$

– **Case - II:** If $\mu = 1$, then we have

$$
\begin{aligned}
\mathsf{round}_{\beta_0}\left(\left|\mathbf{c}_1\begin{pmatrix}\mathbf{h}_{i_k}^\top \\ \mathbf{V}_k\end{pmatrix} - \left(\mathbf{c}_2\mathbf{V}_k + \mathbf{c}_0\mathbf{Z}_k\right)\right|\right) &= \mathsf{round}_{\beta_0}\left(\left|\mathbf{gh}_{i_k}^\top + \widetilde{\mathbf{e}_1} + \widetilde{\mathbf{e}'_1} - \widetilde{\mathbf{e}_0} - \widetilde{\mathbf{e}_2}\right|\right) \\
&= \mathsf{round}_{\beta_0}\left(\left|\mathbf{gh}_{i_k}^\top - \left(\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right)\right|\right)
\end{aligned}
$$

Now, we have the following inequality:

$$
\begin{aligned}
\left|\mathbf{gh}_{i_k}^\top - \left(\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right)\right| &\geq \left|\left|\mathbf{gh}_{i_k}^\top\right| - \left|\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right|\right| \\
&\geq \left|\mathbf{gh}_{i_k}^\top\right| - \left|\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right| \\
&\geq \left|\mathbf{gh}_{i_k}^\top\right| - \beta_0 \quad \left(\because \left|\widetilde{\mathbf{e}_0} + \widetilde{\mathbf{e}_2} - \widetilde{\mathbf{e}_1} - \widetilde{\mathbf{e}'_1}\right| \leq \beta_0\right)
\end{aligned}
$$

From the Proposition 3.2, we have that $\mathbf{g}\mathbf{h}_{i_k}^\top$ is statistically close to uniform over $\mathbb{Z}_p$, and also $\beta_0 \leq \frac{p}{\lambda^{\omega(1)}}$. Thus, with overwhelming probability, we have

$$\mathsf{round}_{\beta_0}\left(\left|\mathbf{g}\mathbf{h}_{i_k}^\top + \widetilde{\mathbf{e}}_1 + \widetilde{\mathbf{e}}_1' - \widetilde{\mathbf{e}}_0 - \widetilde{\mathbf{e}}_2\right|\right) = 1.$$

Therefore, for correct decryption with high probability, it suffices to show that $\left|\widetilde{\mathbf{e}}_0 + \widetilde{\mathbf{e}}_2 - \widetilde{\mathbf{e}}_1 - \widetilde{\mathbf{e}}_1'\right| \leq \beta_0$. This inequality follows from the following computations:

$$\begin{aligned}\left|\widetilde{\mathbf{e}}_0 + \widetilde{\mathbf{e}}_2 - \widetilde{\mathbf{e}}_1 - \widetilde{\mathbf{e}}_1'\right| &\leq |\widetilde{\mathbf{e}}_0| + |\widetilde{\mathbf{e}}_2| + |\widetilde{\mathbf{e}}_1| + \left|\widetilde{\mathbf{e}}_1'\right| \\ &= |\mathbf{e}\mathbf{Z}_k| + |\mathbf{e}\mathbf{K}_2\mathbf{V}_k| + \left|\mathbf{e}\overline{\mathbf{K}_1}\mathbf{h}_{i_k}^\top\right| + \left|\mathbf{e}\underline{\mathbf{K}_1}\mathbf{V}_k\right|\end{aligned}$$

Now, from the Lemma 3.5 and the construction of the Setup algorithm, we have $\|\mathbf{T}\| \leq \sqrt{m}\sigma$, and following the Definition 4.1 and Lemma 4.1, we have

$$\begin{aligned}\|\mathbf{V}_k\| &\leq \mathcal{O}(\|\mathbf{T}\| \cdot m^4 \log p) \leq \mathcal{O}(m^{9/2}\sigma \log p) \\ \|\mathbf{Z}_k\| &\leq \mathcal{O}(\|\mathbf{T}\| \cdot m^7 \log p \log N) \leq \mathcal{O}(m^{15/2}\sigma \log p \log N)\end{aligned}$$

Following the Lemma 3.3, we have $\|\mathbf{e}\| \leq \sqrt{m}\chi'$ and $\|\mathbf{h}_{i_k}\| \leq \sqrt{m}\chi$.

Therefore,

$$\begin{aligned}|\mathbf{e}\mathbf{Z}_k| &\leq \sqrt{m}\chi' \cdot m \cdot \mathcal{O}(m^{15/2}\sigma \log p \log N) = \mathcal{O}(m^9\chi'\sigma \log p \log N), \\ |\mathbf{e}\mathbf{K}_2\mathbf{V}_k| &\leq \sqrt{m}\chi' \cdot m \cdot m \cdot \mathcal{O}(m^{9/2}\sigma \log p) = \mathcal{O}(m^7\chi'\sigma \log p \log N), \\ \left|\mathbf{e}\overline{\mathbf{K}_1}\mathbf{h}_{i_k}^\top\right| &\leq \sqrt{m}\chi' \cdot m \cdot m \cdot \sqrt{m}\chi = m^3\chi\chi', \\ \left|\mathbf{e}\underline{\mathbf{K}_1}\mathbf{V}_k\right| &\leq \sqrt{m}\chi' \cdot m \cdot m \cdot \mathcal{O}(m^{9/2}\sigma \log p) = \mathcal{O}(m^7\chi'\sigma \log p \log N).\end{aligned}$$

From the above calculations,

$$\begin{aligned}\left|\widetilde{\mathbf{e}}_0 + \widetilde{\mathbf{e}}_2 - \widetilde{\mathbf{e}}_1 - \widetilde{\mathbf{e}}_1'\right| &\leq |\mathbf{e}\mathbf{Z}_k| + |\mathbf{e}\mathbf{K}_2\mathbf{V}_k| + \left|\mathbf{e}\overline{\mathbf{K}_1}\mathbf{h}_{i_k}^\top\right| + \left|\mathbf{e}\underline{\mathbf{K}_1}\mathbf{V}_k\right| \\ &\leq \mathcal{O}(m^9\chi'\sigma \log p \log N) + \mathcal{O}(m^7\chi'\sigma \log p \log N) \\ &\qquad + m^3\chi\chi' + \mathcal{O}(m^7\chi'\sigma \log p \log N) \\ &\leq \mathcal{O}(m^9\chi\chi'\sigma \log p \log N) = \beta_0\end{aligned}$$

Hence, the correctness holds with overwhelming high probability.

Theorem 5.2 (Succinctness). *Suppose $m \log p = \mathsf{poly}(\lambda, \log N)$, then the construction 1 is succinct.*

Proof. As per the construction 1, the ciphertext contains three components, $\mathbf{c}_0 \in \mathbb{Z}_p^m$, $\mathbf{c}_1 \in \mathbb{Z}_p^{2m}$ and $\mathbf{c}_2 \in \mathbb{Z}_p^m$, making the total size of the ciphertext to be $\mathcal{O}(m \log p) = \mathsf{poly}(\lambda, \log N)$, fulfilling the requirement for succinctness.

5.3 Proof of Security

This section presents a provable theorem for the semi-static security of our proposed Construction 1.

Theorem 5.3 (Semi-Static Security). *Suppose $\lambda \in \mathbb{N}$ is the security parameter, $N \in \mathbb{N}$ is the total number of users, $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$, the modulus p is prime, $m = \mathcal{O}(n \log p)$, $\chi \geq m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}(m^3 \cdot \log m)$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi \chi' \sigma \log p \log N)$, and $p \geq \beta_0 \cdot \lambda^{\omega(1)}$. Then under the $(2m^2, \sigma)$-succinct LWE assumption with parameters n, m, p, χ', the Construction 1 is semi-statically secure for up to N users.*

Hybrid	Public Parameters	Ciphertext			Public Key	Secret Key
	PP	$\mathbf{c}_0$	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{d}_j^\top$	
$\mathsf{Hyb}_0^{(b)}$	$\boxed{\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}),\ \mathbf{B},\ \mathbf{L}}$	$\boxed{\mathbf{sD} + \mathbf{e}}$	$\boxed{\mathbf{s}(\mathbf{L} \mid \mathbf{A}) + \mathbf{eK}_1 + b\mathbf{g}'}$	$\boxed{\mathbf{s}(\mathbf{C} + \mathbf{A}) + \mathbf{eK}_2}$	$\boxed{\mathbf{L}\mathbf{h}_j^\top + \mathbf{B}\boldsymbol{\gamma}_j^\top}$	$\boxed{\mathbf{h}_j^\top}$
$\mathsf{Hyb}_1^{(b)}$	$\mathbf{T} \xleftarrow{\$} [\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}]^{-1}(\mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$	↓	↓	↓	↓	↓
$\mathsf{Hyb}_2^{(b)}$	$\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$	↓	↓	↓	↓	↓
$\mathsf{Hyb}_3^{(b)}$	↓	↓	↓	↓	$\mathbf{d}_j \xleftarrow{\$} \mathbb{Z}_p^n$	↓
$\mathsf{Hyb}_4^{(b)}$	↓	↓	↓	↓	↓	$\mathbf{h}_j \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m$
$\mathsf{Hyb}_5^{(b)}$	↓	↓	$\mathbf{sL}^* + \mathbf{eK}_1 + b\mathbf{g}'$	↓	↓	↓
$\mathsf{Hyb}_6^{(b)}$	↓	↓	↓	$\mathbf{sA}^* + \mathbf{eK}_2$	↓	↓
$\mathsf{Hyb}_7^{(b)}$	↓	↓	$(\mathbf{sD} + \mathbf{e})\mathbf{K}_1 + b\mathbf{g}'$	$(\mathbf{sD} + \mathbf{e})\mathbf{K}_2$	↓	↓
$\mathsf{Hyb}_8^{(b)}$	↓	$\mathbf{c}_0 \xleftarrow{\$} \mathbb{Z}_p^m$	$\mathbf{c}_0\mathbf{K}_1 + b\mathbf{g}'$	$\mathbf{c}_0\mathbf{K}_2$	↓	↓
$\mathsf{Hyb}_9^{(b)}$	↓	↓	$\mathbf{c}_1 \xleftarrow{\$} \mathbb{Z}_p^{2m}$	$\mathbf{c}_2 \xleftarrow{\$} \mathbb{Z}_p^m$	↓	↓

Table 2: Summary of the hybrid sequence, with the challenge bit $b \in \{0, 1\}$, where $\mathsf{Hyb}_0^{(b)} \approx_s \mathsf{Hyb}_1^{(b)} \approx_s \mathsf{Hyb}_2^{(b)} \approx_s \mathsf{Hyb}_3^{(b)} \approx_s \mathsf{Hyb}_4^{(b)} \approx_s \mathsf{Hyb}_5^{(b)} \approx_s \mathsf{Hyb}_6^{(b)} \approx_s \mathsf{Hyb}_7^{(b)} \approx_c \mathsf{Hyb}_8^{(b)} \approx_s \mathsf{Hyb}_9^{(b)}$. '↓' denotes the same as the previous hybrid. We sample $\mathbf{L}^* \xleftarrow{\$} \mathbb{Z}_p^{n \times 2m}$ in $\mathsf{Hyb}_5^{(b)}$ and $\mathbf{A}^* \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$ in $\mathsf{Hyb}_6^{(b)}$.

Proof. Let $\mathcal{A}$ be any efficient adversary for the semi-static security game and take any $N \leq 2^\lambda$. We will start by defining a sequence of hybrid experiments, which are summarized in Table 2.

- $\mathsf{Hyb}_0^{(b)}$: This will be the semi-static security game against the adversary $\mathcal{A}$ with the challenge bit $b \in \{0, 1\}$. The adversary $\mathcal{A}$ declares a set $\mathcal{S}^* \subseteq [N]$ at the beginning of the game. The challenger samples $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$, $\left(\mathsf{pk}_j, \mathsf{sk}_j\right) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, j\right)$ for each $j \in \mathcal{S}^*$, and sends $\left(\mathsf{PP}, \left\{\mathsf{pk}_j\right\}_{j \in \mathcal{S}^*}\right)$ to the adversary $\mathcal{A}$, where
  - The challenger starts by sampling $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$, $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n \times q}$, where $q = \lceil \log N \rceil + 1$, $(\mathbf{D}, \mathbf{T_D}) \leftarrow \mathsf{TrapGen}\ (1^n, 1^m, p)$, $\mathbf{W} \leftarrow \mathbb{Z}_p^{2m^2 n \times m}$, $\mathbf{T} \leftarrow$

$\mathsf{SamplePre}\left([\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathbf{T_D} \\ 0 \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma\right)$, and then sets the public parameters to be $\mathsf{PP} = (N, \mathsf{pp_{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}), \mathbf{B}, \mathbf{L})$.

- The challenger generates the keys for user $j \in \mathcal{S}^*$ by sampling $\mathbf{h}_j \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi}$, and computing $\mathbf{d}_j^\top = \mathbf{L}\mathbf{h}_j^\top + \mathbf{B}\boldsymbol{\gamma}_j^\top$. The challenger then sets $\mathsf{pk}_j = \mathbf{d}_j$ and $\mathsf{sk}_j = \mathbf{h}_j$.
- In the challenge phase, the adversary sends the challenge set $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq \mathcal{S}^*$ to the challenger, who generates the challenge ciphertext $(\mathsf{CT}_b) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, b\right)$, where the challenger samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi'}$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$, $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$, and $\mathbf{g}' = [\mathbf{g} \mid 0^m]$. The challenger then sets $\mathbf{M} = [\mathbf{M}_1|\mathbf{M}_2|\cdots|\mathbf{M}_\varsigma]$, where $\mathbf{M}_k = \mathbf{d}_{i_k}^\top - \mathbf{B}\boldsymbol{\gamma}_{i_k}^\top$ for each $k \in [\varsigma]$ and computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp_{com}}, \mathbf{M})$. The final ciphertext is

$$\mathsf{CT}_b = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2) = (\mathbf{s}\mathbf{D}+\mathbf{e},\ \mathbf{s}(\mathbf{L} \mid \mathbf{A}) + b\mathbf{g}' + \mathbf{e}\mathbf{K}_1,\ \mathbf{s}(\mathbf{C} + \mathbf{A}) + \mathbf{e}\mathbf{K}_2).$$

- At the end, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ as the output of this experiment.

- $\mathsf{Hyb}_1^{(b)}$: Same as $\mathsf{Hyb}_0^{(b)}$, except for the public parameters, the challenger samples $\mathbf{T} \xleftarrow{\$} [\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}]^{-1}\ (\mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
- $\mathsf{Hyb}_2^{(b)}$: Same as $\mathsf{Hyb}_1^{(b)}$, except the challenger samples $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$.
- $\mathsf{Hyb}_3^{(b)}$: Same as $\mathsf{Hyb}_2^{(b)}$, except the challenger samples $\mathbf{d}_j \xleftarrow{\$} \mathbb{Z}_p^n$ for all $j \in \mathcal{S}^*$.
- $\mathsf{Hyb}_4^{(b)}$: Same as $\mathsf{Hyb}_3^{(b)}$, except the challenger samples $\mathbf{h}_j \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi}$ for all $j \in \mathcal{S}^*$.
- $\mathsf{Hyb}_5^{(b)}$: Same as $\mathsf{Hyb}_4^{(b)}$, except the challenger samples $\mathbf{L}^* \xleftarrow{\$} \mathbb{Z}_p^{n\times 2m}$ and sets $\mathbf{L} \mid \mathbf{A} = \mathbf{L}^*$ in the second component of the ciphertext.
- $\mathsf{Hyb}_6^{(b)}$: Same as $\mathsf{Hyb}_5^{(b)}$, except the challenger now sets $\mathbf{A} = \mathbf{A}^* - \mathbf{C}$, where $\mathbf{A}^* \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$ in the third component of the ciphertext.
- $\mathsf{Hyb}_7^{(b)}$: Same as $\mathsf{Hyb}_6^{(b)}$, except the challenger now sets $\mathbf{L}^* = \mathbf{D}\mathbf{K}_1$ and $\mathbf{A}^* = \mathbf{D}\mathbf{K}_2$. The components in this experiment are as follows:
  - $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$,
  - $\mathbf{T} \xleftarrow{\$} [\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}]^{-1}(\mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$,
  - $\mathbf{c}_0 = \mathbf{s}\mathbf{D} + \mathbf{e}$,
  - $\mathbf{c}_1 = (\mathbf{s}\mathbf{D} + \mathbf{e})\,\mathbf{K}_1 + b\mathbf{g}'$,
  - $\mathbf{c}_2 = (\mathbf{s}\mathbf{D} + \mathbf{e})\,\mathbf{K}_2$.

- $\mathsf{Hyb}_8^{(b)}$: Same as $\mathsf{Hyb}_7^{(b)}$, except the challenger now samples $\mathbf{c}_0 \xleftarrow{\$} \mathbb{Z}_p^m$ and the challenge ciphertext is

$$\mathsf{CT}_b = (\mathbf{c}_0,\ \mathbf{c}_0\mathbf{K}_1 + b\mathbf{g}',\ \mathbf{c}_0\mathbf{K}_2)\,.$$

- $\mathsf{Hyb}_9^{(b)}$: Same as $\mathsf{Hyb}_8^{(b)}$, except the challenger now samples $\mathbf{c}_1 \xleftarrow{\$} \mathbb{Z}_p^{2m}$ and $\mathbf{c}_2 \xleftarrow{\$} \mathbb{Z}_p^m$, and the challenge ciphertext is

$$\mathsf{CT}_b = (\mathbf{c}_0,\ \mathbf{c}_1,\ \mathbf{c}_2)\,.$$

  This experiment is independent of the bit b from the adversary's perspective.

Let the distribution of the output of an execution of $\mathsf{Hyb}_j^{(b)}$, with adversary $\mathcal{A}$, be denoted as $\mathsf{Hyb}_j^{(b)}(\mathcal{A})$. In the following arguments, we show that each subsequent pair of output distributions is indistinguishable. We also give a pictorial view of our sequence of indistinguishability arguments in the Fig. 2.

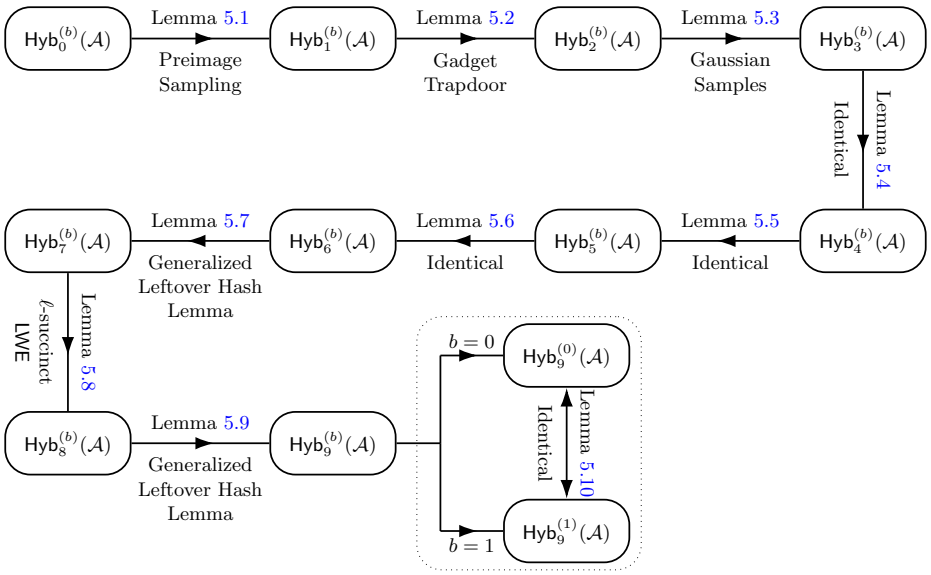


Fig. 2: Hybrid argument for the security proof. Each node $\mathsf{Hyb}_i^{(b)}(\mathcal{A})$ denotes the distribution of the output of adversary $\mathcal{A}$ in the i-th hybrid experiment with challenge bit b. A single arrow '$\longrightarrow$' represents statistical/computational indistinguishability. The final double arrow '$\longleftrightarrow$' indicates that for both $b \in \{0,1\}$, the adversary is in the same experiment. Above each arrow, we write the *lemma* that proves indistinguishability, while below, we state the *underlying assumption*.

Lemma 5.1. *If $n \geq \lambda$, p is prime, $m \geq 3n \log p$, and $\sigma \geq \mathcal{O}(m^3 \log m)$, then for all $b \in \{0,1\}$, we have $\mathsf{Hyb}_0^{(b)}(\mathcal{A}) \approx_s \mathsf{Hyb}_1^{(b)}(\mathcal{A})$.*

Proof. From Lemma 3.1, we know that $\|\mathbf{T_D}\| = 1$. Moreover, if also $\sigma \geq \left(2m^3 + m\right) \|\mathbf{T_D}\| \log(2m^2 n)$, then $\mathbf{T}$ is statistically indistinguishable from a matrix chosen at random from $[\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}]^{-1} (\mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$. Consequently, $\mathsf{Hyb}_0^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_1^{(b)}(\mathcal{A})$ are statistically indistinguishable.

Now, observe that

$$\begin{aligned}
\left(2m^3 + m\right) \|\mathbf{T_D}\| \log(2m^2 n) &= \left(2m^3 + m\right) \log(2m^2 n) \\
&= 2m^3 \log(2m^2 n) + m \log(2m^2 n) \\
&\approx \mathcal{O}(m^3 \log(2m^2 n)) \\
&= \mathcal{O}(m^3 \log m) \\
&\leq \sigma,
\end{aligned}$$

which satisfies the required bound.

Lemma 5.2. *If $n \geq \lambda$, p is prime, and $m \geq 3n \log p$, then $\mathsf{Hyb}_1^{(b)}(\mathcal{A}) \approx_s \mathsf{Hyb}_2^{(b)}(\mathcal{A})$.*

Proof. From the Lemma 3.1, we have that $\mathbf{D}$ is statistically indistinguishable from a matrix chosen uniformly at random from $\mathbb{Z}_p^{n \times m}$ and hence, $\mathsf{Hyb}_1^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_2^{(b)}(\mathcal{A})$ are statistically indistinguishable.

Lemma 5.3. *If $n \geq \lambda$, $p > 2$ is prime, $m \geq 2n \log p$, and $\chi \geq \log m$, then $\mathsf{Hyb}_2^{(b)}(\mathcal{A}) \approx_s \mathsf{Hyb}_3^{(b)}(\mathcal{A})$.*

Proof. Since $\mathbf{B}$ and $\{\boldsymbol{\gamma}_j\}_{j \in \mathcal{S}^*}$ are publicly known, we can equivalently argue the indistinguishability of $\mathbf{d}_j^\top = \mathbf{L}\mathbf{h}_j^\top$ from a uniform random element over $\mathbb{Z}_p^n$. From the Lemma 3.4, we have that the distributions $\left(\mathbf{L}, \mathbf{L}\mathbf{h}_j^\top\right)$ and $\left(\mathbf{L}, \mathbf{d}_j^\top\right)$ are statistically indistinguishable, where $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$, $\mathbf{d}_j \xleftarrow{\$} \mathbb{Z}_p^n$, and $\mathbf{h}_j^\top \leftarrow \mathbf{L}^{-1}\left(\mathbf{d}_j^\top, \chi\right)$. Now, from the construction, it is evident that $\mathbf{h}_j^\top$ is in the set $\mathbf{L}^{-1}\left(\mathbf{d}_j^\top, \chi\right)$, hence the statistical indistinguishability of the output distribution of the experiments follows.

Lemma 5.4. *The distributions of $\mathsf{Hyb}_3^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_4^{(b)}(\mathcal{A})$ are identical.*

Proof. The secret key $\mathbf{h}_j$ for each user $j \in \mathcal{S}^*$ in both experiments is sampled independently from the same Gaussian distribution $\mathcal{D}_{\mathbb{Z},\chi}^m$ with width parameter χ. Therefore, the adversary's point of view remains identical in both experiments.

The experiment $\mathsf{Hyb}_4^{(b)}$ may be omitted, since the adversary $\mathcal{A}$ never interacts with the secret keys of users in $\mathcal{S}^*$. We include it only to make the security proof more explicit.

Lemma 5.5. *The distributions of $\mathsf{Hyb}_4^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_5^{(b)}(\mathcal{A})$ are identical.*

Proof. Since $\mathbf{L}$ and $\mathbf{A}$ are both chosen uniformly at random from $\mathbb{Z}_p^{n\times m}$ by the challenger in the experiment $\mathsf{Hyb}_4^{(b)}$, hence the matrix $\mathbf{L} \mid \mathbf{A}$ can be replaced with a matrix $\mathbf{L}^*$ chosen uniformly at random from $\mathbb{Z}_p^{n\times 2m}$ in the experiment $\mathsf{Hyb}_5^{(b)}$. Therefore, the adversary's point of view remains identical in both experiments.

Lemma 5.6. *The distributions of* $\mathsf{Hyb}_5^{(b)}(\mathcal{A})$ *and* $\mathsf{Hyb}_6^{(b)}(\mathcal{A})$ *are identical.*

Proof. In both experiments, the matrix $\mathbf{A}$ is distributed uniformly at random over $\mathbb{Z}_p^{n\times m}$. Moreover, none of the $\mathbf{d}_j$ depend on $\mathbf{A}$, so the distribution of $\mathbf{A}$ is independent of any commitment matrix $\mathbf{C}$ generated in either experiment. Therefore, the two output distributions are identical.

Lemma 5.7. *If* $n \geq \lambda$, $p > 2$ *is prime,* $m \geq 2n \log p$, *and* $\chi \geq \log m$, *then* $\mathsf{Hyb}_6^{(b)}(\mathcal{A}) \approx_s \mathsf{Hyb}_7^{(b)}(\mathcal{A})$.

Proof. Invoking the Lemma 3.2, with the fixed error vector $\mathbf{e}$ sampled by the challenger in both the experiments $\mathsf{Hyb}_6^{(b)}$ and $\mathsf{Hyb}_7^{(b)}$, we can conclude that $(\mathbf{D}, \mathbf{DK}_1, \mathbf{eK}_1)$ is statistically indistinguishable from $(\mathbf{D}, \mathbf{L}^*, \mathbf{eK}_1)$, where $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{L}^* \xleftarrow{\$} \mathbb{Z}_p^{n\times 2m}$ and $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$, and also $(\mathbf{D}, \mathbf{DK}_2, \mathbf{eK}_2)$ is statistically indistinguishable from $(\mathbf{D}, \mathbf{A}^*, \mathbf{eK}_2)$, where $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{A}^* \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$ and $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$. Hence, the lemma holds.

Lemma 5.8. *If the* $(2m^2, \sigma)$*-succinct LWE assumption holds for the lattice parameters* (n, m, p, χ'), *then for all* $b \in \{0,1\}$, $\mathsf{Hyb}_7^{(b)}(\mathcal{A}) \approx_c \mathsf{Hyb}_8^{(b)}(\mathcal{A})$.

Proof. Suppose, on the contrary, that $\left|\Pr[\mathsf{Hyb}_7^{(b)}(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_8^{(b)}(\mathcal{A}) = 1]\right| \geq \epsilon$ for some non-negligible quantity ϵ, then we can use $\mathcal{A}$ as a subroutine to construct an adversarial algorithm $\mathcal{B}$ for the $(2m^2, \sigma)$-succinct LWE assumption as follows:

1. Given the succinct LWE challenge $(\mathbf{D}, \mathbf{c}_0, \mathbf{W}, \mathbf{T})$, algorithm $\mathcal{B}$ invokes subroutine $\mathcal{A}$, which declares a set $\mathcal{S}^* \subseteq [N]$.
2. Algorithm $\mathcal{B}$ samples $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$ and $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n\times(\lceil \log N\rceil+1)}$. It sets $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}), \mathbf{B}, \mathbf{L})$, and defines $\mathsf{pk}_j = \mathbf{d}_j$ for all $j \in \mathcal{S}^*$, where each $\mathbf{d}_j \xleftarrow{\$} \mathbb{Z}_p^n$.
3. In the challenge phase, $\mathcal{A}$ sends the challenge set $\mathcal{S} = \{i_1, i_2, \ldots, i_\varsigma\} \subseteq \mathcal{S}^*$ to $\mathcal{B}$. Algorithm $\mathcal{B}$ samples $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$ and $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$, sets $\mathbf{M} = \left[\mathbf{d}_{i_1}^\top \mid \mathbf{d}_{i_2}^\top \mid \cdots \mid \mathbf{d}_{i_\varsigma}^\top\right]$, and computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{pmat}}(\mathsf{pp}_{\mathsf{com}}, \mathbf{M})$. Finally, it sets $\mathbf{A} = \mathbf{DK}_2 - \mathbf{C}$.
4. Algorithm $\mathcal{B}$ then constructs the challenge ciphertext

$$\mathsf{CT}_b = (\mathbf{c}_0,\ \mathbf{c}_0\mathbf{K}_1 + b\mathbf{g}',\ \mathbf{c}_0\mathbf{K}_2)\,.$$

5. Finally, $\mathcal{B}$ sends $\left(\mathsf{PP}, \left\{\mathsf{pk}_j\right\}_{j\in\mathcal{S}^*}, \mathsf{CT}_b\right)$ to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs.

As per the definition of the succinct LWE challenge, we have $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{2m^2 n\times m}$, and $\mathbf{T} \leftarrow [\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}]^{-1} (\mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$ and thus, enabling the algorithm $\mathcal{B}$ to perfectly simulate the public parameters and public keys according to the distribution of both $\mathsf{Hyb}_7^{(b)}$ and $\mathsf{Hyb}_8^{(b)}$. To establish the advantage of $\mathcal{B}$, it is sufficient to just consider the challenge ciphertext:

- If the challenger of succinct LWE sets $\mathbf{c}_0 = \mathbf{s}\mathbf{D} + \mathbf{e}$, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^m$, then $\mathcal{B}$ perfectly simulates the experiment $\mathsf{Hyb}_7^{(b)}$.
- If the challenger of succinct LWE sets $\mathbf{c}_0 \xleftarrow{\$} \mathbb{Z}_p^m$, then $\mathcal{B}$ perfectly simulates the experiment $\mathsf{Hyb}_8^{(b)}$.

Thus, the algorithm $\mathcal{B}$ breaks the succinct LWE assumption with the same advantage ϵ, which is a contradiction. Hence, the lemma holds.

Lemma 5.9. *If $n \geq \lambda$, $p > 2$ is prime, $m \geq 2n\log p$, and $\chi \geq \log m$, then $\mathsf{Hyb}_8^{(b)}(\mathcal{A}) \approx_s \mathsf{Hyb}_9^{(b)}(\mathcal{A})$.*

Proof. Consider the matrix $\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix} \in \mathbb{Z}_p^{(n+1)\times m}$, then applying the Lemma 3.2, we get that $\left(\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}, \begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}\mathbf{K}_1\right)$ is statistically indistinguishable from $\left(\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}, \mathbf{U}_1\right)$ and $\left(\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}, \begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}\mathbf{K}_2\right)$ is statistically indistinguishable from $\left(\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}, \mathbf{U}_2\right)$, when $\mathbf{D} \xleftarrow{\$} \mathbb{Z}_p^{n\times m}$, $\mathbf{c}_0 \xleftarrow{\$} \mathbb{Z}_p^m$, $\mathbf{K}_1 \xleftarrow{\$} \{0,1\}^{m\times 2m}$, $\mathbf{K}_2 \xleftarrow{\$} \{0,1\}^{m\times m}$, $\mathbf{U}_1 \xleftarrow{\$} \mathbb{Z}_p^{(n+1)\times 2m}$, and $\mathbf{U}_2 \xleftarrow{\$} \mathbb{Z}_p^{(n+1)\times m}$.

As $\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}\mathbf{K}_1 = \begin{bmatrix}\mathbf{D}\mathbf{K}_1\\ \mathbf{c}_0\mathbf{K}_1\end{bmatrix}$ and $\begin{bmatrix}\mathbf{D}\\ \mathbf{c}_0\end{bmatrix}\mathbf{K}_2 = \begin{bmatrix}\mathbf{D}\mathbf{K}_2\\ \mathbf{c}_0\mathbf{K}_2\end{bmatrix}$, hence, adapting to the above distributions, we get that the distribution $(\mathbf{D}, \mathbf{c}_0, \mathbf{D}\mathbf{K}_1, \mathbf{D}\mathbf{K}_2, \mathbf{c}_0\mathbf{K}_1, \mathbf{c}_0\mathbf{K}_2)$ is statistically indistinguishable from $(\mathbf{D}, \mathbf{c}_0, \mathbf{D}\mathbf{K}_1, \mathbf{D}\mathbf{K}_2, \mathbf{c}_1, \mathbf{c}_2)$, where $\mathbf{c}_1 \xleftarrow{\$} \mathbb{Z}_p^{2m}$ and $\mathbf{c}_2 \xleftarrow{\$} \mathbb{Z}_p^m$. Now, the distribution $(\mathbf{D}, \mathbf{c}_0, \mathbf{D}\mathbf{K}_1, \mathbf{D}\mathbf{K}_2, \mathbf{c}_0\mathbf{K}_1, \mathbf{c}_0\mathbf{K}_2)$ maps to $\mathsf{Hyb}_8^{(b)}$, and the distribution $(\mathbf{D}, \mathbf{c}_0, \mathbf{D}\mathbf{K}_1, \mathbf{D}\mathbf{K}_2, \mathbf{c}_1, \mathbf{c}_2)$ maps to $\mathsf{Hyb}_9^{(b)}$, establishing that the lemma holds.

Lemma 5.10. *The distributions of $\mathsf{Hyb}_9^{(0)}(\mathcal{A})$ and $\mathsf{Hyb}_9^{(1)}(\mathcal{A})$ are identical.*

Proof. As per the construction, the challenger's behavior in $\mathsf{Hyb}_9^{(b)}$ is independent of the challenge bit $b \in \{0,1\}$, so the adversary's view in the two distributions is identical.

Combining Lemmas 5.1 to 5.10, semi-static security of the Construction 1 is ensured through hybrid arguments.

5.4 Parameter Instantiation

Let λ be a security parameter and N be a bound on the number of users. We can instantiate the lattice parameters in Construction 1 to satisfy Theorems 5.1, 5.2, and 5.3 as follows:

- We set the lattice dimension $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$, and $m = \mathcal{O}(n \log p) = n \cdot \mathsf{poly}(\lambda, \log N)$.
- We set the width parameters: $\chi = m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, and $\sigma = \mathcal{O}(m^3 \log m)$.
- Finally, we set the error bound $\beta_0 = \mathcal{O}\left(m^9 \chi \chi' \sigma \log p \log N\right) = \mathsf{poly}(m, \log N)$ and the modulus $p = \beta_0 \cdot \lambda^{\omega(1)} = \mathsf{poly}(m, \log N) \cdot \lambda^{\omega(1)}$.
  In this case, we get that $\log p = \mathcal{O}(\log \lambda + \log N)$.

This setting of parameters corresponds to the succinct LWE instance, where the modulus to noise ratio is $p/\chi' \approx \mathsf{poly}(\log N) \cdot \lambda^{\omega(1)} \approx 2^{\mathcal{O}(n^\epsilon)}$. For this choice of parameters, we obtain a semi-statically secure DBE scheme with the following properties:

- **Public Parameter Size:** Our construction generates public parameters of size $|\mathsf{PP}| = \mathsf{poly}(\lambda, \log N)$.
- **Public Key Size:** The public key for each user is a single vector in $\mathbb{Z}_p^n$, hence, the size of each users's public key as $|\mathsf{pk}| = \mathcal{O}(n \log p) = \mathsf{poly}(\lambda, \log N)$.
- **Secret Key Size:** The secret key of each user is a vector in $\mathbb{Z}_p^m$, hence the size of each user's secret key is $|\mathsf{sk}| = \mathcal{O}(m \log p) = \mathsf{poly}(\lambda, \log N)$.
- **Ciphertext Size:** The ciphertext for a message $\mu \in \{0, 1\}$, contains three vectors respectively in $\mathbb{Z}_p^m$, $\mathbb{Z}_p^{2m}$ and $\mathbb{Z}_p^m$, making the total size of the ciphertext to be $\mathcal{O}(m \log p) = \mathsf{poly}(\lambda, \log N)$.

Corollary 5.1 (Semi-Statically Secure Optimal DBE). *Let λ be a security parameter and N be the number of users. Under the $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption with a sub-exponential modulus-to-noise ratio, there exists a semi-statically secure distributed broadcast encryption scheme in the plain model supporting up to N users, where the size of the public parameters, the size of individual public/secret keys, and the size of the ciphertext is $\mathsf{poly}(\lambda, \log N)$.*

6 Transparent Setup via Decomposed LWE

In the Construction 1, we set $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}), \mathbf{B}, \mathbf{L})$ to be the system public parameters. Since we closely follow the setup of [WW25], where $\mathbf{T}$ is structured as a Gaussian trapdoor associated with $\mathbf{D}$, we are also restricted to a trusted setup regime. But, similar to [WW25], we can also remove this dependence by using an alternate way to realize the matrix commitment [Wee25] from the Decomposed LWE [AMR25].

Let us first recall the analogous matrix commitment scheme from the decomposed LWE assumption, proposed in [Wee25]. Let n, m, p, and χ be lattice parameters such that $m \geq 2n \log p$ and σ be the width parameter. In the following, we denote public parameters for commitment by $\mathsf{pp}^d_{\mathsf{com}} = (\widehat{\mathbf{D}}, \mathbf{W}, \mathbf{R})$, where $\mathbf{W} \in \mathbb{Z}_p^{n \times 2m^3}$, $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m \times 2m^3}$, and $\widehat{\mathbf{D}} = \mathbf{W}\left(\mathbf{I}_{2m^2} \otimes \mathbf{R}\right) + \mathsf{vec}\left(\mathbf{I}_{2m^2}\right)^\top \otimes \mathbf{G} \in \mathbb{Z}_p^{n \times 4m^5}$.

Lemma 6.1 (Matrix Commitment [Wee25, WW25]). *There exist three efficient and deterministic algorithms* $\left(\mathsf{Com}^{\mathsf{mat}}, \mathsf{Ver}^{\mathsf{mat}}, \mathsf{Open}^{\mathsf{mat}}\right)$ *with the following syntax:*

- $(\mathbf{C}) \leftarrow \mathsf{Com}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, \mathbf{M}\right)$*: Given* $\mathbf{M} \in \mathbb{Z}_p^{n \times L}$*, outputs* $\mathbf{C} \in \mathbb{Z}_p^{n \times m}$.
- $(\mathbf{V}) \leftarrow \mathsf{Ver}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, 1^L\right)$*: Given* L*, outputs* $\mathbf{V} \in \mathbb{Z}_p^{m \times L}$.
- $(\mathbf{Z}) \leftarrow \mathsf{Open}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, \mathbf{M}\right)$*: Given* $\mathbf{M} \in \mathbb{Z}_p^{n \times L}$*, outputs* $\mathbf{Z} \in \mathbb{Z}_p^{4m^5 \times L}$.

For all $L \in \mathbb{N}$*, all* $\mathbf{M} \in \mathbb{Z}_p^{n \times L}$*, all* $\mathsf{pp}^d_{\mathsf{com}} = (\widehat{\mathbf{D}}, \mathbf{W}, \mathbf{R})$*, and setting* $\mathbf{C} = \mathsf{Com}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, \mathbf{M}\right)$*,* $\mathbf{V} = \mathsf{Ver}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, 1^L\right)$ *and* $\mathbf{Z} = \mathsf{Open}^{\mathsf{mat}}\left(\mathsf{pp}^d_{\mathsf{com}}, \mathbf{M}\right)$*, we have*

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} - \mathbf{D} \cdot \mathbf{Z},$$

with $\|\mathbf{V}\| \leq \mathcal{O}\left(\|\mathbf{R}\| \cdot m^4 \log p\right)$ *and* $\|\mathbf{Z}\| \leq \mathcal{O}\left(\|\mathbf{R}\| \cdot m^7 \log p \log L\right)$*. The running time of all three algorithms is bounded above by* $L \cdot \log L \cdot \mathsf{poly}(m)$.

Accordingly, substituting this alternate definition of matrix commitment into Construction 1 yields a DBE scheme based on the decomposed LWE assumption. This substitution leads to the following modifications in our scheme:

- **Transparent Setup:** The commitment parameters in the above instantiation can be described by $\mathbf{W} \leftarrow \mathbb{Z}_p^{n \times 2m^3}$ and $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m \times 2m^3}$. The matrix $\mathbf{D}$ is replaced by $\widehat{\mathbf{D}}$. The other two matrices, $\mathbf{L}$ and $\mathbf{B}$, in Construction 1 are already uniformly random, thereby yielding a *transparent* setup process.
- **Security:** The semi-static security of Construction 1 follows from the $2m^2$-succinct LWE assumption, which can be restated to assert that

  $$\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{s}\mathbf{D} + \mathbf{e}\right) \approx_c \left(\mathsf{pp}_{\mathsf{com}}, \mathbf{c}\right),$$

  where $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$ is the succinct LWE instance of dimension $2m^2$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^m$, and $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^m$.

  In the substituted construction, the security requirement is slightly modified to

  $$\left(\mathsf{pp}^d_{\mathsf{com}}, \mathbf{s}\widehat{\mathbf{D}} + \mathbf{e}\right) \overset{?}{\approx}_c \left(\mathsf{pp}_{\mathsf{com}}, \mathbf{c}\right),$$

where $\mathsf{pp}^d_{\mathsf{com}} = \left(\widehat{\mathbf{D}}, \mathbf{W}, \mathbf{R}\right)$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{n \times 2m^3}$, $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m \times 2m^3}$, $\widehat{\mathbf{D}} = \mathbf{W}\left(\mathbf{I}_{2m^2} \otimes \mathbf{R}\right) + \mathsf{vec}\left(\mathbf{I}_{2m^2}\right)^\top \otimes \mathbf{G}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$, $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^{4m^5}$, and $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^{4m^5}$. The required indistinguishability then follows directly from the decomposed LWE assumption with dimension $2m^2$ and width σ.

- **Larger Parameters:** A potential drawback of this substitution is the increase in parameter sizes, since $\widehat{\mathbf{D}} \in \mathbb{Z}_p^{n \times 4m^5}$ has width $4m^5$, compared to $\mathbf{D} \in \mathbb{Z}_p^{n \times m}$ of width m. At first glance, this appears to cause a significant blow-up in size. However, it is important to note that this growth remains polynomial, as $m = \mathsf{poly}(\lambda, \log N)$.

7 Adaptively-Secure DBE in Random Oracle Model

This section presents the upgradation of our semi-statically secure DBE scheme from Section 5 to an adaptively-secure one in the Random Oracle Model (ROM). Our approach adapts the techniques of [GW09,KMW23]. For clarity, in this section, we consider the (one-time) trust setup setting. Similar to the modifications of Section 6 in Construction 1, the transparent setup can also be used in the following Construction 2.

7.1 Construction of Unbounded Optimal DBE from Succinct LWE

We consider $N \leq 2^\lambda$ and propose a transformed construction of our DBE scheme, which achieves optimal parameter sizes and is proven adaptively-secure in the ROM in the subsequent sections.

Construction 2. Let λ be the security parameter, N denote a bound on the number of users, (n, m, p, χ') be the LWE parameters, χ and σ denote Gaussian width parameters. We can then construct an adaptively-secure DBE scheme $\prod^{\mathsf{Ad\text{-}R}}_{\mathsf{DBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in the ROM as follows:

- $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, N\right)$: Given the security parameter λ and the bound N on the total number of users in the system as inputs, the setup algorithm proceeds as follows:
  1. For ease of exposition, the set $[2N]$ is indexed using a pair $(i, \beta) \in [N] \times \{0, 1\}$.
  2. Sets $q = \lceil \log N \rceil + 1$, and samples $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_p^{n \times q}$ and $\mathbf{L} \xleftarrow{\$} \mathbb{Z}_p^{n \times m}$.
  3. Samples the pair $(\mathbf{D}, \mathbf{T_D}) \leftarrow \mathsf{TrapGen}\left(1^n, 1^m, p\right)$, and chooses $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{2m^2 n \times m}$.
  4. Runs $\mathbf{T} \leftarrow \mathsf{SamplePre}\left(\left[\mathbf{I}_{2m^2} \otimes \mathbf{D} \mid \mathbf{W}\right], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathbf{T_D} \\ 0 \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma\right)$, and sets $\mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T})$.

5. Chooses a cryptographically secure hash function $\mathcal{H} : \{0,1\}^{\mathcal{O}(\lambda)} \times [N] \to \{0,1\}$.

The algorithm finally outputs the public parameters as: $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}}, \mathbf{B}, \mathbf{L}, \mathcal{H})$.

- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\,(\mathsf{PP}, i)$: On input the public parameters PP and an index $i \in [N]$, the key-generation algorithm proceeds as follows:

  1. Samples $\mathbf{h}_{i,0}, \mathbf{h}_{i,1} \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi}$.
  2. Sets $\mathbf{d}^\top_{i,0} = \mathbf{L} \cdot \mathbf{h}^\top_{i,0} + \mathbf{B} \cdot \boldsymbol{\gamma}^\top_i$, and $\mathbf{d}^\top_{i,1} = \mathbf{L} \cdot \mathbf{h}^\top_{i,1} + \mathbf{B} \cdot \boldsymbol{\gamma}^\top_i$.
  3. Samples a bit $s_i \in \{0,1\}$ uniformly at random.

  The algorithm finally sets the public key as $\mathsf{pk}_i = (\mathbf{d}_{i,0}, \mathbf{d}_{i,1})$, and the secret key as $\mathsf{sk}_i = (s_i, \mathbf{h}_{i,s_i})$.

- $(\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mu\right)$: Given the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ together with their corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$, and a one-bit message μ, the broadcaster executes the encryption algorithm as follows:

  1. Samples $\mathbf{v} \xleftarrow{\$} \{0,1\}^{\mathcal{O}(\lambda)}$ and computes $t_{i_k} \leftarrow \mathcal{H}(\mathbf{v}, i_k)$ for all $k \in [\varsigma]$.
  2. Sets $\mathcal{S}_0 = \{(i_k, t_{i_k})\}_{k\in[\varsigma]}$ and $\mathcal{S}_1 = \{(i_k, 1-t_{i_k})\}_{k\in[\varsigma]}$.
  3. Samples $\mathbf{s}_0, \mathbf{s}_1 \xleftarrow{\$} \mathbb{Z}^n_p$, $\mathbf{e}_0, \mathbf{e}_1 \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi'}$, $\mathbf{A}_0, \mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}^{n\times m}_p$, $\mathbf{K}_{1,0}, \mathbf{K}_{1,1} \xleftarrow{\$} \{0,1\}^{m\times 2m}$ and $\mathbf{K}_{2,0}, \mathbf{K}_{2,1} \xleftarrow{\$} \{0,1\}^{m\times m}$ to set $\mathbf{g}' = [\mathbf{g} \mid 0^m]$.
  4. For all $k \in [\varsigma]$, sets $\mathbf{M}_{k,0} = \left[\mathbf{d}^\top_{i_k,t_{i_k}} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_k}\right]$ and $\mathbf{M}_{k,1} = \left[\mathbf{d}^\top_{i_k,1-t_{i_k}} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_k}\right]$, and then forms the matrices $\mathbf{M}^0 = [\mathbf{M}_{1,0} \mid \mathbf{M}_{2,0} \mid \cdots \mid \mathbf{M}_{\varsigma,0}]$ and $\mathbf{M}^1 = [\mathbf{M}_{1,1} \mid \mathbf{M}_{2,1} \mid \cdots \mid \mathbf{M}_{\varsigma,1}]$.
  5. Computes $\mathbf{C}_0 \leftarrow \mathsf{Com}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}^0\right)$ and $\mathbf{C}_1 \leftarrow \mathsf{Com}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}^1\right)$.
  6. Sets the ciphertext components as follows:

$$\mathsf{CT}^0 = \left(\overbrace{\mathbf{s}_0\mathbf{D} + \mathbf{e}_0}^{\mathbf{c}^0_0},\ \overbrace{\mathbf{s}_0(\mathbf{L} \mid \mathbf{A}_0) + \mu\mathbf{g}' + \mathbf{e}_0\mathbf{K}_{1,0}}^{\mathbf{c}^0_1},\ \overbrace{\mathbf{s}_0(\mathbf{C}_0 + \mathbf{A}_0) + \mathbf{e}_0\mathbf{K}_{2,0}}^{\mathbf{c}^0_2}\right),$$

$$\mathsf{CT}^1 = \left(\underbrace{\mathbf{s}_1\mathbf{D} + \mathbf{e}_1}_{\mathbf{c}^1_0},\ \underbrace{\mathbf{s}_1(\mathbf{L} \mid \mathbf{A}_1) + \mu\mathbf{g}' + \mathbf{e}_1\mathbf{K}_{1,1}}_{\mathbf{c}^1_1},\ \underbrace{\mathbf{s}_1(\mathbf{C}_1 + \mathbf{A}_1) + \mathbf{e}_1\mathbf{K}_{2,1}}_{\mathbf{c}^1_2}\right).$$

  At the end, it sets the final ciphertext as: $\mathsf{CT} = \left(\mathbf{v}, \mathsf{CT}^0, \mathsf{CT}^1\right)$.

- $(\mu) \leftarrow \mathsf{Dec}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mathsf{CT}, \left(j, \mathsf{sk}_j\right)\right)$: Taking as input the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ along with their corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$, a valid ciphertext CT, and the secret key sk_j of the user indexed by j such that $j \in \mathcal{S}$, *i.e.*, $j = i_k$ for some $k \in [\varsigma]$, the user j executes the decryption algorithm as follows:
  1. For all $k \in [\varsigma]$, computes $t_{i_k} \leftarrow \mathcal{H}(\mathbf{v}, i_k)$.
  2. Finds $k \in [\varsigma]$ such that $j = i_k$, defines $\beta = t_{i_k} \oplus s_{i_k}$, and parses the component CT^β.
  3. Sets $\mathbf{M} = \left[\mathbf{d}^\top_{i_1, t_{i_1}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_1} \mid \mathbf{d}^\top_{i_2, t_{i_2}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_2} \mid \cdots \mid \mathbf{d}^\top_{i_\varsigma, t_{i_\varsigma}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_\varsigma}\right]$.
  4. Samples $\mathbf{V}_k \leftarrow \mathsf{Ver}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, 1^\varsigma, k\right)$ and $\mathbf{Z}_k \leftarrow \mathsf{Open}^{\mathsf{pmat}}\left(\mathsf{pp}_{\mathsf{com}}, \mathbf{M}, k\right)$.

  The decryptor finally outputs

$$\mu' = \mathsf{round}_{\beta_0}\left(\left|\mathbf{c}_1^\beta \begin{pmatrix}\mathbf{h}^\top_{i_k, s_{i_k}} \\ \mathbf{V}_k\end{pmatrix} - \left(\mathbf{c}_2^\beta \mathbf{V}_k + \mathbf{c}_0^\beta \mathbf{Z}_k\right)\right|\right),$$

  where β_0 is the error bound.

7.2 Proof of Correctness and Succinctness

This section presents the correctness and succinctness of the Construction 2 through formal theorems.

Theorem 7.1 (Correctness). *Suppose $\lambda \in \mathbb{N}$ is the security parameter, $N \in \mathbb{N}$ denote the total number of users, $n \geq \lambda$, the modulus p is prime, $m = \mathcal{O}(n \log p)$, $\chi \geq m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}\left(m^3 \cdot \log m\right)$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi\chi'\sigma \log p \log N)$, and $p \geq \beta_0 \cdot \lambda^{\omega(1)}$. Assuming that there exists a user indexed by j such that $j \in \mathcal{S}$, the Construction 2 satisfies the protocol correctness.*

Proof. Let the system public parameters $\mathsf{PP} = (N, \mathsf{pp}_{\mathsf{com}} = (\mathbf{D}, \mathbf{W}, \mathbf{T}), \mathbf{B}, \mathbf{L}, \mathcal{H})$ be generated by the $\mathsf{Setup}\left(1^\lambda, N\right)$ algorithm. Also, for any index $i \in [N]$, consider $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, i\right)$. Take any set $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ and the corresponding collection of public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$. Now, choose a message $\mu \in \{0,1\}$ and let $\left(\mathbf{v}, \mathsf{CT}^0, \mathsf{CT}^1\right) = (\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mu\right)$. Finally, set $t_{i_k} = \mathcal{H}(\mathbf{v}, i_k)$ for each $k \in [\varsigma]$.

Since $j \in \mathcal{S}$, therefore, there exists a $k \in [\varsigma]$ such that $j = i_k$.

Case-I ($\beta = 0$): We have $s_{i_k} = t_{i_k}$, and therefore

$$\begin{aligned}\mathbf{M} &= \left[\mathbf{d}^\top_{i_1, t_{i_1}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_1} \mid \mathbf{d}^\top_{i_2, t_{i_2}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_2} \mid \cdots \mid \mathbf{d}^\top_{i_\varsigma, t_{i_\varsigma}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_\varsigma}\right] \\ &= \left[\mathbf{d}^\top_{i_1, t_{i_1}} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_1} \mid \mathbf{d}^\top_{i_2, t_{i_2}} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_2} \mid \cdots \mid \mathbf{d}^\top_{i_\varsigma, t_{i_\varsigma}} - \mathbf{B}\boldsymbol{\gamma}^\top_{i_\varsigma}\right] \\ &= \mathbf{M}^0.\end{aligned}$$

The parsed component from the ciphertext is given by

$$\mathsf{CT}^{\beta} = \mathsf{CT}^{0} = (\mathbf{s}_0\mathbf{D} + \mathbf{e}_0,\ \mathbf{s}_0(\mathbf{L} \mid \mathbf{A}_0) + \mu\mathbf{g}' + \mathbf{e}_0\mathbf{K}_{1,0},\ \mathbf{s}_0(\mathbf{C}_0 + \mathbf{A}_0) + \mathbf{e}_0\mathbf{K}_{2,0})\,.$$

The parsed component from the secret key is

$$\mathbf{h}_{j,s_j} \ = \ \mathbf{h}_{i_k,s_{i_k}} \ = \ \mathbf{h}_{i_k,t_{i_k}}.$$

The component from the public key used for decryption is

$$\mathbf{d}^{\top}_{i_k,t_{i_k}} \ = \ \mathbf{L}\mathbf{h}^{\top}_{i_k,t_{i_k}} + \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_k} \ = \ \mathbf{L}\mathbf{h}^{\top}_{i_k,s_{i_k}} + \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_k}.$$

Thus, the correctness in this case follows from the above derivations in conjunction with Theorem 5.1.

Case-II ($\beta = 1$): We have $s_{i_k} = 1 - t_{i_k}$, and therefore

$$\begin{aligned}
\mathbf{M} \ &= \ \left[\mathbf{d}^{\top}_{i_1,t_{i_1}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_1} \mid \mathbf{d}^{\top}_{i_2,t_{i_2}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_2} \mid \cdots \mid \mathbf{d}^{\top}_{i_\varsigma,t_{i_\varsigma}\oplus\beta} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_\varsigma}\right] \\
&= \ \left[\mathbf{d}^{\top}_{i_1,t_{i_1}\oplus 1} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_1} \mid \mathbf{d}^{\top}_{i_2,t_{i_2}\oplus 1} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_2} \mid \cdots \mid \mathbf{d}^{\top}_{i_\varsigma,t_{i_\varsigma}\oplus 1} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_\varsigma}\right] \\
&= \ \left[\mathbf{d}^{\top}_{i_1,1-t_{i_1}} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_1} \mid \mathbf{d}^{\top}_{i_2,1-t_{i_2}} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_2} \mid \cdots \mid \mathbf{d}^{\top}_{i_\varsigma,1-t_{i_\varsigma}} - \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_\varsigma}\right] \\
&= \ \mathbf{M}^{1}.
\end{aligned}$$

The parsed component from the ciphertext is given by

$$\mathsf{CT}^{\beta} = \mathsf{CT}^{1} = (\mathbf{s}_1\mathbf{D} + \mathbf{e}_1,\ \mathbf{s}_1(\mathbf{L} \mid \mathbf{A}_1) + \mu\mathbf{g}' + \mathbf{e}_1\mathbf{K}_{1,1},\ \mathbf{s}_1(\mathbf{C}_1 + \mathbf{A}_1) + \mathbf{e}_1\mathbf{K}_{2,1})\,.$$

The parsed component from the secret key is

$$\mathbf{h}_{j,s_j} \ = \ \mathbf{h}_{i_k,s_{i_k}} \ = \ \mathbf{h}_{i_k,1-t_{i_k}}.$$

The component from the public key used for decryption is

$$\mathbf{d}^{\top}_{i_k,1-t_{i_k}} \ = \ \mathbf{L}\mathbf{h}^{\top}_{i_k,1-t_{i_k}} + \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_k} \ = \ \mathbf{L}\mathbf{h}^{\top}_{i_k,s_{i_k}} + \mathbf{B}\boldsymbol{\gamma}^{\top}_{i_k}.$$

Likewise, the correctness in this case also follows from Theorem 5.1.

Therefore, the Construction 2 is correct.

Theorem 7.2 (Succinctness). *Suppose $m \log p = \mathsf{poly}(\lambda, \log N)$, then the construction 2 is succinct.*

Proof. As per the Construction 2, the size of the ciphertext is twice the size of the ciphertext obtained in Construction 1 along with a $\mathcal{O}(\lambda)$ size bit vector, yielding a total size of $2 \cdot \mathsf{poly}(\lambda, \log N) + \mathcal{O}(\lambda) = \mathsf{poly}(\lambda, \log N)$, thereby satisfying the requirement for succinctness.

7.3 Proof of Security

Based on the proof techniques of [GPV08, KMW23], we formally establish the adaptive security of our Construction 2 in the ROM in the following theorem.

Theorem 7.3 (Adaptive Security in ROM). *Suppose $\lambda \in \mathbb{N}$ is the security parameter, $N \leq 2^\lambda$ denote the total number of users, $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$, the modulus p is a prime, $m = \mathcal{O}(n \log p)$, $\chi > m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}\left(m^3 \cdot \log m\right)$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi \chi' \sigma \log p \log N)$, and $p \geq \beta_0 \cdot \lambda^{\omega(1)}$. Then under the $\left(2m^2, \sigma\right)$-succinct LWE assumption with parameters n, m, p, χ', the Construction 2 is adaptively-secure for up to N users.*

Proof. Since we are adapting the transformation techniques presented in [GW09, KMW23], if the underlying DBE scheme (*i.e.*, Construction 1) is semi-statically secure for up to $2N$ users (from Theorem 5.3), then the transformed scheme (*i.e.*, Construction 2) is adaptively-secure for up to N users.

7.4 Parameter Instantiation

Let λ be the security parameter and N denote a bound on the number of users. We instantiate the lattice parameters in Construction 2 to satisfy Theorems 7.1, 7.2, and 7.3, as in Section 5.4, where the lattice dimension $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$, $m = n \cdot \mathsf{poly}(\lambda, \log N)$, the width parameters $\chi = m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, and $\sigma = \mathcal{O}(m^3 \log m)$, and the modulus $p = \mathsf{poly}(m, \log N) \cdot \lambda^{\omega(1)}$.

This setting of parameters corresponds to the succinct LWE instance where the modulus-to-noise ratio is $2^{\mathcal{O}(n^\epsilon)}$. With this choice of parameters, we obtain an adaptively-secure DBE scheme in the ROM with the following properties:

- **Public Parameter Size:** Our construction generates public parameters of size $|\mathsf{PP}| = \mathsf{poly}(\lambda, \log N)$.
- **Public Key Size:** The public key for each user is a pair of vectors in $\mathbb{Z}_p^n$, hence, the size of each users's public key is $|\mathsf{pk}| = \mathcal{O}(n \log p) = \mathsf{poly}(\lambda, \log N)$.
- **Secret Key Size:** The secret key of each user is a bit in $\{0,1\}$ and a vector in $\mathbb{Z}_p^m$, hence the size of each user's secret key is $|\mathsf{sk}| = \mathcal{O}(m \log p) + 1 = \mathsf{poly}(\lambda, \log N)$.
- **Ciphertext Size:** The ciphertext for a message $\mu \in \{0,1\}$, contains a vector in $\{0,1\}^{\mathcal{O}(\lambda)}$, four vectors in $\mathbb{Z}_p^m$, and two vectors in $\mathbb{Z}_p^{2m}$, so the total size of the ciphertext is $\mathcal{O}(m \log p) + \mathcal{O}(\lambda) = \mathsf{poly}(\lambda, \log N)$.

Corollary 7.1 (Adaptively-Secure Optimal DBE in ROM). *Let λ be a security parameter and N denote the number of users. Under the $\mathsf{poly}(\lambda, \log N)$-succinct LWE assumption with a sub-exponential modulus-to-noise ratio, there exists an adaptively-secure distributed broadcast encryption scheme in the random oracle model supporting up to N users where the size of the public parameters, the size of individual public/secret keys, and the ciphertext are $\mathsf{poly}(\lambda, \log N)$.*

8 Adaptively-Secure DBE in Plain Model

This section presents the upgradation of our semi-statically secure DBE scheme from Section 5 to an adaptively-secure one using publicly sampleable projective PRGs in the plain model. Our approach adapts the techniques of [HWW25]. From Lemma 3.8, in the context of feasibility, we are restricted to a priori bounded user setting since the LWE instance of the publicly-sampleable projective PRG implies linear-sized public parameters.

8.1 Construction of Almost Optimal DBE from Succinct LWE

Construction 3. Let λ denote the security parameter, N represent a bound on the number of users, and (n, m, p, χ') be the LWE parameters. Furthermore, let χ, σ and ρ be Gaussian width parameters, and let B be a parameter such that $p > 4B + 2$. We construct an adaptively-secure DBE scheme $\prod_{\mathsf{DBE}}^{\mathsf{Ad\text{-}P}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ in the plain model as follows:

- $(\mathsf{PP}) \leftarrow \mathsf{Setup}\left(1^\lambda, 1^N\right)$: Taking the security parameter λ and the bound N on the total number of users as input, the algorithm proceeds as follows:

  1. Executes the first four steps in the same way as the setup algorithm in Construction 2.
  2. Samples $\mathbf{P} \xleftarrow{\$} \mathbb{Z}_p^{n \times qm}$ and $\mathbf{p}, \mathbf{r}, \mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_N \xleftarrow{\$} \mathbb{Z}_p^n$.
  3. Defines the set
  $$\mathsf{NR}_{p,B} := \mathbb{Z}_p \setminus \left\{ \left[\left\lfloor \frac{p}{4} \right\rfloor - B, \left\lfloor \frac{p}{4} \right\rfloor + B \right] \cup \left[- \left\lfloor \frac{p}{4} \right\rfloor - B, - \left\lfloor \frac{p}{4} \right\rfloor + B \right] \right\}.$$
  4. For all $i \in [N]$, samples $y_i \xleftarrow{\$} \mathsf{NR}_{p,B}$, $\widehat{\mathbf{e}}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi''}^{qm}$, and $\widehat{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\rho}$.
  5. Sets up the Indicator function $I_{\mathbf{u}} : \{0,1\}^q \to \{0,1\}$ for any $\mathbf{u} \in \{0,1\}^q$ as follows:
  $$I_{\mathbf{u}}(\mathbf{v}) = \begin{cases} 0 & \mathbf{v} \neq \mathbf{u} \\ 1 & \mathbf{v} = \mathbf{u} \end{cases}$$
  6. Computes $\mathbf{P}_i \leftarrow \mathsf{EvalF}(\mathbf{P}, I_{\boldsymbol{\gamma}_i})$ for all $i \in [N]$.
  7. For all $i \in [N]$, computes $\mathbf{x}_i = \mathbf{k}_i \left(\mathbf{P} - \boldsymbol{\gamma}_i \otimes \mathbf{G}\right) + \widehat{\mathbf{e}}_i$ and $z_i = (\mathbf{r} + \mathbf{k}_i)\, \mathbf{P}_i \mathbf{G}^{-1}(\mathbf{p}^\top) + \widehat{e}_i + y_i$.
  8. Sets $\mathsf{pp}_{\mathsf{prg}} = \left(\mathbf{P}, \mathbf{p}, \{\mathbf{x}_i, z_i\}_{i \in [N]}\right)$ and $\delta = (\mathbf{r}, \mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_N)$.

  The algorithm finally outputs the public parameters as: $\mathsf{PP} = (\mathsf{pp}_{\mathsf{com}}, \mathsf{pp}_{\mathsf{prg}}, \delta, \mathbf{B}, \mathbf{L})$.

- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\,(\mathsf{PP}, i)$: Given the public parameters PP and an index $i \in [N]$, the key-generation algorithm proceeds in the same way as in Construction 2.

- $(\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mu\right)$: Given as input the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ with their corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$, and a one-bit message μ, the broadcaster executes the encryption algorithm as follows:

  1. For all $k \in [\varsigma]$, computes $\mathbf{P}_{i_k} \leftarrow \mathsf{EvalF}(\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}})$ and $\mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \leftarrow \mathsf{EvalFX}(\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l})$.

  2. Samples $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^n$, and sets $t_{i_k} = \lfloor y'_{i_k} \rceil$ for all $k \in [\varsigma]$, where

$$y'_{i_k} \;=\; z_{i_k} - \mathbf{v}\mathbf{P}_{i_k}\mathbf{G}^{-1}(\mathbf{p}^\top) + \sum_{k\neq l=1}^{\varsigma} \mathbf{x}_{i_l}\mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}}\mathbf{G}^{-1}(\mathbf{p}^\top),$$

     and

$$\lfloor y'_{i_k} \rceil \;=\; \begin{cases} 0 & |y'_{i_k}| \leq \frac{p}{4} \\ 1 & \text{otherwise.} \end{cases}$$

  3. The algorithm now follows the encryption algorithm of the Construction 2.

  At last, it sets the final ciphertext, same as the Construction 2, as $\mathsf{CT} = \left(\mathbf{v}, \mathsf{CT}^0, \mathsf{CT}^1\right)$.

- $(\mu) \leftarrow \mathsf{Dec}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k\in[\varsigma]}, \mathsf{CT}, (j, \mathsf{sk}_j)\right)$: Taking as input the public parameters PP, a set of indices $\mathcal{S} = \{i_1, i_2, \cdots, i_\varsigma\} \subseteq [N]$ together with their corresponding public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k\in[\varsigma]}$, a valid ciphertext CT, and the secret key sk_j of the user indexed by j such that $j \in \mathcal{S}$, *i.e.*, $j = i_k$ for some $k \in [\varsigma]$, the user j executes the decryption algorithm as follows:

  1. For all $k \in [\varsigma]$, computes $\mathbf{P}_{i_k} \leftarrow \mathsf{EvalF}(\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}})$ and $\mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \leftarrow \mathsf{EvalFX}(\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l})$.

  2. Computes $t_{i_k} = \lfloor y'_{i_k} \rceil$ for all $k \in [\varsigma]$, where

$$y'_{i_k} \;=\; z_{i_k} - \left(\mathbf{r} + \sum_{l=1}^{\varsigma} \mathbf{k}_{i_l}\right)\mathbf{P}_{i_k}\mathbf{G}^{-1}(\mathbf{p}^\top) + \sum_{k\neq l=1}^{\varsigma} \mathbf{x}_{i_l}\mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}}\mathbf{G}^{-1}(\mathbf{p}^\top).$$

  3. Finds $k \in [\varsigma]$ such that $j = i_k$, sets $\beta = t_{i_k} \oplus s_{i_k}$ and then proceeds in the same way as the decryption algorithm in Construction 2.

8.2 Proof of Correctness and Succinctness

We consider a priori-bounded user setting and present the correctness and succinctness of the Construction 3 in the form of the following theorems.

Theorem 8.1 (Correctness). *Suppose $\lambda \in \mathbb{N}$ be the security parameter, $N = N(\lambda)$ represent a polynomial, $n \geq \lambda$, the modulus p is a prime, $m = \mathcal{O}(n \log p)$, $\chi \geq m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}\left(m^3 \cdot \log m\right)$, $\chi'' = \mathsf{poly}(\lambda)$, $\rho = 2^{\lambda} \cdot \chi''$ $m^{\mathcal{O}(\log \log N)}$ $N \log N$, $B \geq 2\sqrt{\lambda}\rho$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi \chi' \sigma \log p \log N)$, and $p \geq 2^{\lambda} NB \cdot \beta_0$. Assuming that there exists a user, indexed by j such that $j \in \mathcal{S}$, the Construction 3 satisfies the correctness of the protocol.*

Proof. Let $\mathsf{PP} = \left(\mathsf{pp}_{\mathsf{com}} = (\mathbf{D},\ \mathbf{W},\ \mathbf{T}),\ \mathsf{pp}_{\mathsf{prg}} = \left(\mathbf{P}, \mathbf{p}, \{\mathbf{x}_i, z_i\}_{i \in [N]}\right),\ \mathbf{B},\ \mathbf{L},\ \delta = \left(\mathbf{r}, \{\mathbf{k}_i\}_{i \in [N]}\right)\right)$ denote public parameters of the system, generated from the $\mathsf{Setup}\left(1^{\lambda}, 1^{N}\right)$ algorithm. Also, for any index $i \in [N]$, consider $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}\left(\mathsf{PP}, i\right)$. Take any set $\mathcal{S} = \{i_1, i_2, \cdots, i_{\varsigma}\} \subseteq [N]$ and the corresponding collection of public keys $\left\{\mathsf{pk}_{i_k}\right\}_{k \in [\varsigma]}$. Now choose a message $\mu \in \{0,1\}$ and let $\left(\mathbf{v}, \mathsf{CT}^0, \mathsf{CT}^1\right) = (\mathsf{CT}) \leftarrow \mathsf{Enc}\left(\mathsf{PP}, \mathcal{S}, \left\{\left(i_k, \mathsf{pk}_{i_k}\right)\right\}_{k \in [\varsigma]}, \mu\right)$.

Since $j \in \mathcal{S}$, therefore, there exists a $k \in [\varsigma]$ such that $j = i_k$. For all $l \in [\varsigma] \setminus \{k\}$, we have

$$
\begin{aligned}
& \mathbf{x}_{i_l} \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top}) \\
&= \left(\mathbf{k}_{i_l} \left(\mathbf{P} - \boldsymbol{\gamma}_{i_l} \otimes \mathbf{G}\right) + \widehat{\mathbf{e}_{i_l}}\right) \cdot \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top}) \\
&= \mathbf{k}_{i_l} \left(\mathbf{P} - \boldsymbol{\gamma}_{i_l} \otimes \mathbf{G}\right) \cdot \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top}) + \underbrace{\widehat{\mathbf{e}_{i_l}} \cdot \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top})}_{\widehat{\mathbf{e}^*_{i_l}}} \\
&= \mathbf{k}_{i_l} \mathbf{P}_{i_l} \mathbf{G}^{-1}(\mathbf{p}^{\top}) + \widehat{\mathbf{e}^*_{i_l}}.
\end{aligned}
$$

Now, from the property of sampling indistinguishability, we have that $\mathbf{v} = \mathbf{r} + \sum_{l=1}^{\varsigma} \mathbf{k}_{i_l}$. Hence, the algorithm computes the following:

$$
\begin{aligned}
y'_{i_k} &= z_{i_k} - \mathbf{v} \mathbf{P}_{i_k} \mathbf{G}^{-1}(\mathbf{p}^{\top}) + \sum_{k \neq l = 1}^{\varsigma} \mathbf{x}_{i_l} \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top}) \\
&= z_{i_k} - \left(\mathbf{r} + \sum_{l=1}^{\varsigma} \mathbf{k}_{i_l}\right) \mathbf{P}_{i_k} \mathbf{G}^{-1}(\mathbf{p}^{\top}) + \sum_{k \neq l = 1}^{\varsigma} \mathbf{x}_{i_l} \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top}) \\
&= \left(\mathbf{r} + \mathbf{k}_{i_k}\right) \mathbf{P}_{i_k} \mathbf{G}^{-1}(\mathbf{p}^{\top}) + \widehat{e_{i_k}} + y_{i_k} - \left(\mathbf{r} + \sum_{l=1}^{\varsigma} \mathbf{k}_{i_l}\right) \mathbf{P}_{i_k} \mathbf{G}^{-1}(\mathbf{p}^{\top}) \\
&\qquad + \sum_{k \neq l = 1}^{\varsigma} \mathbf{x}_{i_l} \mathbf{H}_{\mathbf{P}, I_{\boldsymbol{\gamma}_{i_k}}, \boldsymbol{\gamma}_{i_l}} \mathbf{G}^{-1}(\mathbf{p}^{\top})
\end{aligned}
$$

$$= (\mathbf{r} + \mathbf{k}_{i_k})\,\mathbf{P}_{i_k}\mathbf{G}^{-1}(\mathbf{p}^\top) + \widehat{e_{i_k}} + y_{i_k} - \left(\mathbf{r} + \sum_{l=1}^{\varsigma} \mathbf{k}_{i_l}\right)\mathbf{P}_{i_k}\mathbf{G}^{-1}(\mathbf{p}^\top) + \sum_{k\neq l=1}^{\varsigma} \left(\mathbf{k}_{i_l}\mathbf{P}_{i_l}\mathbf{G}^{-1}(\mathbf{p}^\top) + \widehat{\mathbf{e}^*_{i_l}}\right)$$

$$= y_{i_k} + \widehat{e_{i_k}} + \sum_{k\neq l=1}^{\varsigma} \widehat{\mathbf{e}^*_{i_l}}.$$

Now,

$$\begin{aligned}\left|\widehat{e_{i_k}} + \sum_{k\neq l=1}^{\varsigma} \widehat{\mathbf{e}^*_{i_l}}\right| &\leq |\widehat{e_{i_k}}| + \left|\sum_{k\neq l=1}^{\varsigma} \widehat{\mathbf{e}^*_{i_l}}\right| \\ &\leq \sqrt{\lambda}\rho + \varsigma \cdot q \cdot m^{\mathcal{O}(\log\log N)}\chi''\sqrt{\lambda} \\ &\leq \sqrt{\lambda}\rho + N \cdot \log N \cdot m^{\mathcal{O}(\log\log N)}\chi''\sqrt{\lambda} \\ &\leq \sqrt{\lambda}\rho + \sqrt{\lambda}\rho \\ &= 2\sqrt{\lambda}\rho \;=\; B.\end{aligned}$$

Since $y'_{i_k} \in \mathsf{NR}_{p,B}$, we have $y'_{i_k} = y_{i_k}$. The above calculation is independent of the index i_k, yielding $y'_{i_l} = y_{i_l}$ for all $l \in [\varsigma]$, and hence for the entire set $\mathcal{S}$. Finally, setting $\beta = t_{i_k} \oplus s_{i_k}$ ensures the correctness of Theorems 5.1 and 7.1.

Theorem 8.2 (Succinctness). *Suppose $m \log p = \mathsf{poly}(\lambda, \log N)$, then the Construction 3 is succinct.*

Proof. According to Construction 3, the size of the ciphertext is twice the size of the ciphertext obtained in Construction 1, together with a projected seed $\mathbf{v}$ in $\mathbb{Z}_p^n$, which makes the size $2 \cdot \mathcal{O}(m \log p) + \mathcal{O}(n \log p) \leq \mathcal{O}(m \log p) = \mathsf{poly}(\lambda, \log N)$, thus fulfilling the requirement of succinctness.

8.3 Proof of Security

By the techniques of [HWW25], in the following theorem, we prove that our Construction 3 achieves adaptive security in the plain model.

Theorem 8.3 (Adaptive Security in Plain Model). *Suppose $\lambda \in \mathbb{N}$ is the security parameter, $N = N(\lambda)$ denote a polynomial, $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$, the modulus p is a prime, $m = \mathcal{O}(n \log p)$, $\chi \geq m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\sigma = \mathcal{O}\left(m^3 \cdot \log m\right)$, $\chi'' = \mathsf{poly}(\lambda)$, $\rho = 2^\lambda \cdot \chi'' \; m^{\mathcal{O}(\log\log N)}$ $N \log N$, $B \geq 2\sqrt{\lambda}\rho$, the error bound $\beta_0 = \mathcal{O}(m^9 \chi\chi'\sigma \log p \log N)$, and $p \geq 2^\lambda NB \cdot \beta_0$. Then, under the $(2m^2, \sigma)$-succinct LWE assumption with parameters n, m, p, χ' and the $\mathsf{LWE}_{n,m,p,\chi''}$ assumption, the Construction 3 is adaptively-secure for up to N users.*

Proof. Since we are adapting the transformation techniques presented in the work of [HWW25], if the underlying DBE scheme (*i.e.*, Construction 1) is semi-statically secure for up to $2N$ users (from Theorem 5.3), and if there exists a publicly-sampleable projective PRG satisfying correctness, sampling indistinguishability and succinctness, then the transformed scheme (*i.e.*, Construction 3) is adaptively-secure for up to N users.

8.4 Parameter Instantiation

Let λ be a security parameter and N denote a bound on the number of users. We instantiate the lattice parameters in Construction 3 to satisfy Theorems 8.1, 8.2, and 8.3 as follows:

- We set the lattice dimension $n = (\lambda \log N)^{1/\epsilon}$ for some $0 < \epsilon < 1$ and $m = \mathcal{O}(n \log p) = n \cdot \mathsf{poly}(\lambda, \log N)$.
- We also set the width parameters as $\chi = m \log n$, $\chi' = \mathsf{poly}(\lambda, \log N)$, $\chi'' = \mathsf{poly}(\lambda)$, $\sigma = \mathcal{O}(m^3 \log m)$, and $\rho = 2^{\lambda} \cdot N \cdot \mathsf{poly}(m, \log N)$.
- We finally set the bounds $\beta_0 = \mathsf{poly}(m, \log N)$ and $B = 2\sqrt{\lambda}\rho$, and define the modulus $p = 2^{\lambda} NB \cdot \beta_0 = 2^{\mathcal{O}(\lambda)} \cdot N^2 \cdot \mathsf{poly}(m, \log N)$. In this case, it follows that $\log p = \mathcal{O}(\lambda + \log N)$.

This setting of parameters corresponds to the succinct LWE instance, where the modulus to noise ratio is $p/\chi' \approx \mathsf{poly}(\lambda, \log N) \cdot 2^{\mathcal{O}(\lambda)} \approx 2^{\mathcal{O}(n^{\epsilon})}$ and the LWE instance with the modulus to noise ratio $p/\chi'' \approx \mathsf{poly}(m, \log N) \cdot 2^{\mathcal{O}(\lambda)} \approx 2^{\mathcal{O}(n^{\epsilon})}$. Hence, with this choice of parameters, we obtain an adaptively-secure DBE scheme with the following properties:

- **Public Parameter Size:** Our construction generates public parameters of size $|\mathsf{PP}| = N \cdot \mathsf{poly}(\lambda, \log N)$.
- **Public Key Size:** The public key for each user is a pair of vectors in $\mathbb{Z}_p^n$, therefore, the size of the public key of each user is $|\mathsf{pk}| = \mathcal{O}(n \log p) = \mathsf{poly}(\lambda, \log N)$.
- **Secret Key Size:** The secret key of each user is a bit in $\{0, 1\}$ and a vector in $\mathbb{Z}_p^m$, hence the size of each user's secret key is given by $|\mathsf{sk}| = \mathcal{O}(m \log p) + 1 = \mathsf{poly}(\lambda, \log N)$.
- **Ciphertext Size:** The ciphertext for a message $\mu \in \{0, 1\}$ contains a projected seed vector in $\mathbb{Z}_p^n$, four vectors in $\mathbb{Z}_p^m$, and two vectors in $\mathbb{Z}_p^{2m}$, making the total size of the ciphertext $\mathcal{O}(m \log p) + \mathcal{O}(n \log p) = \mathsf{poly}(\lambda, \log N)$.

Corollary 8.1 (Adaptively-Secure Almost Optimal DBE in Plain Model). *Let λ be the security parameter and N denote the total number of users. Under the $\mathsf{poly}(\lambda, \log N)$-succinct LWE and LWE assumptions with a sub-exponential modulus-to-noise ratio, there exists an adaptively-secure distributed broadcast encryption scheme in the plain model supporting up to N users where the size of the public parameters is $N \cdot \mathsf{poly}(\lambda, \log N)$, and the size of individual public/secret keys and the ciphertext are $\mathsf{poly}(\lambda, \log N)$.*

Acknowledgements

This work is supported in part by the Anusandhan National Research Foundation (ANRF), Department of Science and Technology (DST), Government of India, under File No. EEQ/2023/000164 and the Information Security Education and Awareness (ISEA) Project Phase-III initiatives of the Ministry of Electronics and Information Technology (MeitY) under Grant No. F.No. L-14017/1/2022-HRD.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 553–572. Springer, 2010. https://doi.org/10.1007/978-3-642-13190-5_28.

[ABI+23] Benny Applebaum, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, and Vinod Vaikuntanathan. Succinct computational secret sharing. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1553–1566, 2023. https://doi.org/10.1145/3564246.3585127.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996. https://dl.acm.org/doi/pdf/10.1145/237814.237838.

[AMR25] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Key-homomorphic computations for RAM: fully succinct randomised encodings and more. In *Annual International Cryptology Conference*, pages 236–268. Springer, 2025. https://doi.org/10.1007/978-3-032-01881-6_8.

[AMYY25] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Evasive LWE: Attacks, Variants & Obfustopia. *Cryptology ePrint Archive*, 2025. https://eprint.iacr.org/2025/375.pdf.

[AT24] Nuttapong Attrapadung and Junichi Tomida. A modular approach to Registered ABE for unbounded predicates. In *Annual International Cryptology Conference*, pages 280–316. Springer, 2024. https://doi.org/10.1007/978-3-031-68382-4_9.

[AY20] Shweta Agrawal and Shota Yamada. Optimal Broadcast Encryption from Pairings and LWE. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I 39*, pages 13–43. Springer, 2020. https://doi.org/10.1007/978-3-030-45721-1_2.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 533–556. Springer, 2014. https://doi.org/10.1007/978-3-642-55220-5_30.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001. https://doi.org/10.1007/3-540-44647-8_1.

[BLM+24] Pedro Branco, Russell WF Lai, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Ivy KY Woo. Traitor tracing without trusted authority from registered functional encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 33–66. Springer, 2024. https://doi.org/10.1007/978-981-96-0891-1_2.

[BW06] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 211–220, 2006. https://doi.org/10.1145/1180405.1180432.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty Key Exchange, Efficient Traitor Tracing, and More from Indistinguishability Obfuscation. In *Advances in Cryptology – CRYPTO 2014*, pages 480–499, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44371-2_27.

[CES21] Kelong Cong, Karim Eldefrawy, and Nigel P Smart. Optimizing registration based encryption. In *IMA International Conference on Cryptography and Coding*, pages 129–157. Springer, 2021. https://doi.org/10.1007/978-3-030-92641-0_7.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25:601–639, 2012. https://doi.org/10.1007/s00145-011-9105-2.

[CHW25] Jeffrey Champion, Yao-Ching Hsieh, and David J Wu. Registered ABE and adaptively-secure broadcast encryption from succinct LWE. In *Annual International Cryptology Conference*, pages 3–34. Springer, 2025. https://doi.org/10.1007/978-3-032-01881-6_1.

[CW24] Jeffrey Champion and David J Wu. Distributed broadcast encryption from lattices. In *Theory of Cryptography Conference*, pages 156–189. Springer, 2024. https://doi.org/10.1007/978-3-031-78020-2_6.

[DKL+23] Nico Döttling, Dimitris Kolonelos, Russell WF Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 417–446. Springer, 2023. https://doi.org/10.1007/978-3-031-30620-4_14.

[DPY25] Pratish Datta, Tapas Pal, and Shota Yamada. Registered FE beyond predicates:(attribute-based) linear functions and more. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 65–104. Springer, 2025. https://doi.org/10.1007/978-981-96-0875-1_3.

[FFM+23] Danilo Francati, Daniele Friolo, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Daniele Venturi. Registered (inner-product) functional encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 98–133. Springer, 2023. https://doi.org/10.1007/978-981-99-8733-7_4.

[FKP23] Dario Fiore, Dimitris Kolonelos, and Paola de Perthuis. Cuckoo commitments: registration-based encryption and key-value map commitments for

large spaces. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 166–200. Springer, 2023. https://doi.org/10.1007/978-981-99-8733-7_6.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In *Annual International Cryptology Conference*, pages 480–491. Springer, 1993. https://doi.org/10.1007/3-540-48329-2_40.

[FWW23] Cody Freitag, Brent Waters, and David J Wu. How to use (plain) witness encryption: registered ABE, flexible broadcast, and more. In *Annual International Cryptology Conference*, pages 498–531. Springer, 2023. https://doi.org/10.1007/978-3-031-38551-3_16.

[GHM+19] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. Registration-based encryption from standard assumptions. In *IACR international workshop on public key cryptography*, pages 63–93. Springer, 2019. https://doi.org/10.1007/978-3-030-17259-6_3.

[GHMR18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: removing private-key generator from IBE. In *Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part I 16*, pages 689–718. Springer, 2018. https://doi.org/10.1007/978-3-030-03807-6_25.

[GKMR23] Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. Efficient registration-based encryption. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1065–1079, 2023. https://doi.org/10.1145/3576915.3616596.

[GKPW24] Sanjam Garg, Dimitris Kolonelos, Guru-Vamsi Policharla, and Mingyuan Wang. Threshold encryption with silent setup. In *Annual International Cryptology Conference*, pages 352–386. Springer, 2024. https://doi.org/10.1007/978-3-031-68394-7_12.

[GLWW23] Rachit Garg, George Lu, Brent Waters, and David J Wu. Realizing flexible broadcast encryption: how to broadcast to a public-key directory. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1093–1107, 2023. https://doi.org/10.1145/3576915.3623168.

[GLWW24] Rachit Garg, George Lu, Brent Waters, and David J Wu. Reducing the CRS size in registered ABE systems. In *Annual International Cryptology Conference*, pages 143–177. Springer, 2024. https://doi.org/10.1007/978-3-031-68382-4_5.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008. https://doi.org/10.1145/1374376.1374407.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual cryptology conference*, pages 75–92. Springer, 2013. https://doi.org/10.1007/978-3-642-40041-4_5.

[GV20] Rishab Goyal and Satyanarayana Vusirikala. Verifiable registration-based encryption. In *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part I 40*, pages 621–651. Springer, 2020. https://doi.org/10.1007/978-3-030-56784-2_21.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 171–188. Springer, 2009. https://doi.org/10.1007/978-3-642-01001-9_10.

[HLWW23] Susan Hohenberger, George Lu, Brent Waters, and David J Wu. Registered attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 511–542. Springer, 2023. https://doi.org/10.1007/978-3-031-30620-4_17.

[HWW25] Yao-Ching Hsieh, Brent Waters, and David J Wu. A generic approach to adaptively-secure broadcast encryption in the plain model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 336–365. Springer, 2025. https://doi.org/10.1007/978-3-031-91131-6_12.

[KMW23] Dimitris Kolonelos, Giulio Malavolta, and Hoeteck Wee. Distributed broadcast encryption from bilinear groups. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 407–441. Springer, 2023. https://doi.org/10.1007/978-981-99-8733-7_13.

[MKMS22] Jose Maria Bermudo Mera, Angshuman Karmakar, Tilen Marc, and Azam Soleimanian. Efficient lattice-based inner-product functional encryption. In *IACR International Conference on Public-Key Cryptography*, pages 163–193. Springer, 2022. https://doi.org/10.1007/978-3-030-97131-1_6.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012. https://doi.org/10.1007/978-3-642-29011-4_41.

[MR09] Daniele Micciancio and Oded Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. https://doi.org/10.1007/978-3-540-88702-7_5.

[PPS12] Duong Hieu Phan, David Pointcheval, and Mario Strefler. Decentralized dynamic broadcast encryption. In *International Conference on Security and Cryptography for Networks*, pages 166–183. Springer, 2012. https://doi.org/10.1007/978-3-642-32928-9_10.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *Annual International Cryptology Conference*, pages 535–559. Springer, 2022. https://doi.org/10.1007/978-3-031-15802-5_19.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 195–221. Springer, 2022. https://doi.org/10.1007/978-3-031-22963-3_7.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 217–241. Springer, 2022. https://doi.org/10.1007/978-3-031-07085-3_8.

[Wee24] Hoeteck Wee. Circuit ABE with poly (depth, λ)-sized ciphertexts and keys from lattices. In *Annual International Cryptology Conference*, pages 178–209. Springer, 2024. https://doi.org/10.1007/978-3-031-68382-4_6.

[Wee25] Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In *Annual International Conference on the Theory and*

Applications of Cryptographic Techniques, pages 34–62. Springer, 2025. https://doi.org/10.1007/978-3-031-91131-6_2.

[WQZDF10] Qianhong Wu, Bo Qin, Lei Zhang, and Josep Domingo-Ferrer. Ad hoc broadcast encryption. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 741–743, 2010. https://doi.org/10.1145/1866307.1866416.

[WW23] Hoeteck Wee and David J Wu. Succinct vector, polynomial, and functional commitments from lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 385–416. Springer, 2023. https://doi.org/10.1007/978-3-031-30620-4_13.

[WW25] Hoeteck Wee and David J Wu. Unbounded distributed broadcast encryption and registered ABE from succinct LWE. In *Annual International Cryptology Conference*, pages 204–235. Springer, 2025. https://doi.org/10.1007/978-3-032-01881-6_7.

[ZZGQ23] Ziqi Zhu, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered ABE via predicate encodings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 66–97. Springer, 2023. https://doi.org/10.1007/978-981-99-8733-7_3.