Distributed Broadcast Encryption from Lattices

Jeffrey Champion
UT Austin
jchampion@utexas.edu

David J. Wu
UT Austin
dwu4@cs.utexas.edu

Abstract

A broadcast encryption scheme allows a user to encrypt a message to N recipients with a ciphertext whose size scales sublinearly with N. While broadcast encryption enables succinct encrypted broadcasts, it also introduces a strong trust assumption and a single point of failure; namely, there is a central authority who generates the decryption keys for all users in the system. Distributed broadcast encryption offers an appealing alternative where there is a one-time (trusted) setup process that generates a set of public parameters. Thereafter, users can independently generate their own public keys and post them to a public-key directory. Moreover, anyone can broadcast an encrypted message to any subset of user public keys with a ciphertext whose size scales sublinearly with the size of the broadcast set. Unlike traditional broadcast encryption, there are no long-term secrets in distributed broadcast encryption and users can join the system at any time (by posting their public key to the public-key directory).

Previously, distributed broadcast encryption schemes were known from standard pairing-based assumptions or from powerful tools like indistinguishability obfuscation or witness encryption. In this work, we provide the first distributed broadcast encryption scheme from a falsifiable lattice assumption. Specifically, we rely on the ℓ-succinct learning with errors (LWE) assumption introduced by Wee (CRYPTO 2024). Previously, the only lattice-based candidate for distributed broadcast encryption goes through general-purpose witness encryption, which in turn is only known from the *private-coin* evasive LWE assumption, a strong and *non-falsifiable* lattice assumption. Along the way, we also describe a more direct construction of broadcast encryption from lattices.

1 Introduction

Suppose a user wants to encrypt a message to a set of users S. With vanilla public-key encryption, the encrypter would separately encrypt the message under each user's public key, and then broadcast the set of $|S|$ ciphertexts. Each user can read the message by decrypting their respective ciphertext in the broadcast. In this case, the size of the encrypted broadcast scales *linearly* with the size of the set $|S|$. Broadcast encryption [FN93] provides an elegant approach for achieving *succinct* encrypted broadcasts. With broadcast encryption, the encrypter can encrypt a message to an arbitrary set of S users with a ciphertext whose length scales *sublinearly* with $|S|$. However, broadcast encryption achieves this savings at a cost of introducing a central *trusted* authority that generates the public parameters for the scheme as well as each user's individual decryption key. Broadcast encryption thus has built-in key escrow, and indeed, if the central authority is ever compromised, then the attacker learns the secret keys for every single user in the system. This is in direct contrast to the setting with public-key encryption where each user generates their own cryptographic keys. A natural question is whether we can achieve the efficiency advantages of broadcast encryption *without* relying on a trusted centralized authority.

Distributed broadcast encryption. To circumvent the key escrow problem implicit in broadcast encryption, several works have introduced the notion of *distributed broadcast encryption* [WQZDF10, BZ14]. Distributed broadcast encryption is a hybrid between public-key encryption and broadcast encryption. Like the setting of public-key encryption, users in distributed broadcast encryption generate their own public/secret key-pairs and then post their public keys to a public-key directory (i.e., a public bulletin board). Anyone can encrypt a message to an arbitrary collection of public keys with a ciphertext whose size scales sublinearly with the size of the broadcast set (much like in traditional broadcast encryption). Note that in distributed broadcast encryption, we assume the encrypter and the decrypter know the set of public keys associated with a ciphertext (similar to how in broadcast encryption, both the encrypter and the decrypter know the set of users associated with the broadcast). While there is no trusted authority in distributed broadcast encryption, we do allow for a one-time trusted sampling of a set of public parameters. The trusted setup only needs to be performed once (e.g., using multiparty computation) and the same set of public parameters can be shared across multiple schemes. There are no long-term secrets in the scheme following the initial setup process. Thus, distributed broadcast encryption (and its generalizations) provide an elegant way to combine the decentralized, trustless nature of public-key encryption with the efficiency benefits of broadcast encryption.

To date, distributed broadcast encryption is known from indistinguishability obfuscation [BZ14], witness encryption [FWW23], as well as assumptions over bilinear groups [WQZDF10, KMW23, GKPW24]. The work of [FWW23] also shows how to generic construct a distributed broadcast encryption scheme from a registered attribute-based encryption (ABE) scheme; several recent works have shown how to construct registered ABE from pairing-based assumptions [HLWW23, ZZGQ23, GLWW24, AT24]. Among these constructions, the only one from plausibly post-quantum assumptions is the one based on witness encryption, which can be constructed using lattice assumptions [Tsa22, VWW22]—specifically, the evasive learning with errors (LWE) assumption [Wee22, Tsa22]. However, evasive LWE is a strong *non-falsifiable* lattice assumption, and moreover, existing constructions of witness encryption rely on a *private-coin* version of evasive LWE. As noted in [VWW22], there are (heuristic) obfuscation-based counter-examples for the general version of private-coin evasive LWE, so the status of private-coin evasive LWE remains unsettled. A natural goal then is to obtain simpler and more direct constructions of distributed broadcast encryption from (preferably falsifiable) lattice assumptions. An even better objective would be to obtain distributed broadcast encryption from the plain LWE assumption, but to date, even the simpler notion of centralized broadcast encryption from LWE remains a long-standing open problem. Existing centralized broadcast encryption schemes from lattice assumptions either lack a security proof [BV22], or rely on new lattice assumptions such as (public-coin) evasive LWE [Wee22] or ℓ-succinct LWE [Wee24].

This work. In this work, we give the first distributed broadcast encryption scheme from a *falsifiable* lattice assumption. Specifically, we rely on the ℓ-succinct LWE assumption recently introduced by Wee [Wee24] for constructing broadcast encryption and succinct attribute-based encryption. The ℓ-succinct LWE assumption essentially asserts that $(\mathbf{A}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal)$ is pseudorandom even given a trapdoor for the related matrix $\mathbf{V} = [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]$ where $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$, $\mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{\ell n\times m}$, and χ is an error distribution. We provide more details in Section 1.1. The ℓ-succinct LWE assumption is a falsifiable assumption and is implied by (public-coin) evasive LWE (in combination with LWE). Variants of this assumption (adapted to the setting of short integer solutions) have also been used in recent constructions of succinct functional commitments [ACL+22, WW23a, CLM23, BCFL23, WW23b, FMN23]. We summarize our results with the following informal theorem and provide a comparison to previous distributed broadcast encryption schemes in Table 1.

Scheme	Assumption	$\lvert pp\rvert$	$\lvert pk\rvert$	$\lvert sk\rvert$	$\lvert ct\rvert$	TP	PQ
Generic	public-key encryption	–	1	1	$\lvert S\rvert$	✓	✓
Generic	registration-based encryption	1	1	1	$\lvert S\rvert$	✓	✓
[WQZDF10]	bilinear Diffie-Hellman exponent	N	N^2	N	1	✓	✗
[BZ14]	$i\mathcal{O}$ + one-way function	–	1	1	1	✓	✗
[FWW23]*	witness encryption + LWE	1	1	1	1	✓	✓
[KMW23]	bilinear Diffie-Hellman exponent	N	N	1	1	✗	✗
[KMW23]	k-Lin (pairing group)	N^2	N	1	1	✗	✗
[GKPW24]	generic bilinear group	N	N	1	1	✗	✗
This work	ℓ-succinct LWE	N^2	N	1	1	✗	✓

* The work of [FWW23] also describe a generic approach for constructing distributed broadcast encryption from any registered attribute-based encryption (ABE) scheme. A number of recent works have shown how to construct registered ABE from bilinear maps [HLWW23, ZZGQ23, GLWW24, AT24]. Since these generic instantiations do not improve upon other the other bilinear-map-based constructions already shown in the table, we omit these for simplicity of comparison.

Table 1: Comparison with existing distributed broadcast encryption schemes. For each scheme, we report the size of the public parameters pp, the user public/secret key-pair $(\mathsf{pk}, \mathsf{sk})$, and the ciphertext ct as a function of the number of users N, and the size of the broadcast set $|S|$. For simplicity of comparison, we suppress $\mathsf{poly}(\lambda, \log N)$ factors, where λ is the security parameter. For each scheme, we also indicate whether the public parameters pp (if required) can be generated using a *transparent* setup procedure (TP), and whether it is (plausibly) post-quantum secure (PQ). The first two rows describe generic *non-succinct* approaches of using public-key encryption (PKE) or registration-based encryption (RBE) [GHMR18] to separately encrypt to each user in the broadcast set. We write $i\mathcal{O}$ to denote indistinguishability obfuscation [BGI+01, GGH+13]. The parameter ℓ in ℓ-succinct LWE must satisfy $\ell \geq N \cdot O(\lambda \log N)$.

Theorem 1.1 (Informal)**.** *Let λ be a security parameter and N be a bound on the number of users. Then, under the ℓ-succinct LWE assumption (with $\ell \geq N \cdot O(\lambda \log N)$), there exists a distributed broadcast encryption scheme that supports up to N users with the following properties:*

- *The public parameters consist of a structured string of size $N^2 \cdot \mathsf{poly}(\lambda, \log N)$.*
- *Each user's public key has size $O(N\lambda \log^2 N)$ and secret key has size $O(\lambda \log^2 N)$.*
- *An encryption to a set of $S \subseteq [N]$ users has size $O(\lambda \log^2 N)$.*
- *Encryption and decryption with respect to a set S take time $|S| \cdot \mathsf{poly}(\lambda, \log N)$. Moreover, if the set S is known in advance, we can precompute a set-dependent encryption key pk_S; encrypting to the set S then requires $\mathsf{poly}(\lambda, \log N)$ time. Similarly, each user $i \in S$ can also precompute a set-dependent decryption key $\mathsf{sk}_{S,i}$; decrypting a ciphertext associated with S then requires $\mathsf{poly}(\lambda, \log N)$ time.*

Open problems. Our work gives the first distributed broadcast encryption scheme from a falsifiable lattice assumption. Our scheme has a quadratic-size CRS. An interesting open problem is to obtain a distributed broadcast encryption scheme with a linear (or even sublinear-size) CRS from a falsifiable lattice assumption. Schemes with linear-size public parameters are known from bilinear maps (under either the

bilinear Diffie-Hellman exponent assumption or in the generic bilinear group model) [KMW23, GKPW24]. Another interesting question is to construct a distributed broadcast encryption scheme that is able to support an a priori unbounded number of users. Currently, this is only known from witness encryption and indistinguishability obfuscation. Note that if we alternatively impose a bound on the size of the broadcast set, then the transformation of [GLWW23] can be used to obtain a scheme that supports an arbitrary number of users (but where each ciphertext can only target a bounded subset of users).

On the ℓ-succinct LWE assumption. Security of our lattice-based distributed broadcast encryption scheme relies on the ℓ-succinct LWE assumption recently introduced by Wee [Wee24]. Prior to this work, distributed broadcast encryption was known from witness encryption [FWW23], which can be built from evasive LWE [Tsa22, VWW22]. Since both approaches rely on non-standard lattice assumptions, it is natural to ask whether it is worthwhile to study constructions from ℓ-succinct LWE if we already have one from evasive LWE. We provide a brief discussion here and refer to [Wee24, §1.4] for additional perspectives.

First, unlike evasive LWE, the ℓ-succinct LWE assumption is falsifiable. The ℓ-succinct LWE assumption is also implied by (public-coin) evasive LWE together with plain LWE (c.f., [Wee24, §6.2]), so formally, ℓ-succinct LWE is a *weaker* assumption than evasive LWE. On the other hand, evasive LWE is non-falsifiable, and must be carefully-formulated to avoid counter-examples. In particular, there are obfuscation-based counter-examples for the general version of *private-coin* evasive LWE [Wee22, VWW22]. Existing constructions of witness encryption based on evasive LWE [Tsa22, VWW22] all rely on private-coin versions of evasive LWE. Note that the known counter-examples for private-coin evasive LWE pertain only to the most general version of the assumption, and not to the specific distributions needed by [Tsa22, VWW22].

A second advantage of the ℓ-succinct LWE assumption over evasive LWE is that it is "instance-independent." We reduce to the same assumption irrespective of the adversary. In contrast, when reducing security to evasive LWE, the matrices in the pre- and post-conditions are typically functions of the adversary (specifically, the queries that the adversary makes). Formally, this is captured by defining a sampling algorithm based on the adversary. So even though the evasive LWE post-condition itself is a falsifiable assumption, there is typically a *different* post-condition for each adversary. As such, when analyzing security, we are relying on a family of computational assumptions (one for each adversary) as opposed to a single instance-independent assumption (that applies to all adversaries). Since ℓ-succinct LWE is falsifiable and instance-independent, the ℓ-succinct LWE assumption provides a concrete target for cryptanalysis, especially compared to evasive LWE.

There has also recently been a proliferation of new (falsifiable) lattice assumptions. Most of these correspond to some variant of the short integer solutions (SIS) problem or the LWE problem with hints [ACL+22, WW23b, CLM23, BCFL23, WW23a, FMN23, AFLN24]; see [Alb24] for a survey and comparison. Essentially, these assumptions assert that SIS or LWE is hard with respect to a matrix $\mathbf{A}$ even given some structured preimage $\mathbf{A}^{-1}(\mathbf{P})$ for some matrix $\mathbf{P}$. Among these, the ℓ-succinct SIS assumption is weaker (up to polynomial losses in the parameters) than assumptions like $\mathsf{BASIS}_{\mathsf{struct}}$ or k-R-ISIS assumptions considered in many of the aforementioned works. From this perspective, we believe ℓ-succinct SIS and ℓ-succinct LWE to be an appealing assumption to use when studying new lattice-based constructions.

Finally, if we compare our distributed broadcast encryption scheme directly to the one based on witness encryption, we obtain a much more direct construction (conceptually similar to classic pairing-based broadcast encryption schemes [BGW05, GW09]). For instance, the witness encryption approach makes heavy non-black-box use of cryptographic objects (specifically, the witness encryption scheme is applied to a function-binding hash function, which itself relies on leveled homomorphic encryption to construct). In contrast, our approach directly realizes the broadcast functionality and does not need any kind of

homomorphic encryption machinery. We believe this to be a significant conceptual benefit of our approach.

1.1 Technical Overview

In this section, we provide a high-level overview of our approach for constructing distributed broadcast encryption from lattices.

Notation. We write $D_{\mathbb{Z},\sigma}$ to denote the discrete Gaussian distribution over $\mathbb{Z}$ with width parameter $\sigma > 0$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a target vector $\mathbf{t} \in \mathbb{Z}_q^n$, we write $\mathbf{A}^{-1}(\mathbf{t})$ to denote a random variable $\mathbf{x} \leftarrow D_{\mathbb{Z},\sigma}^m$ conditioned on $\mathbf{Ax} = \mathbf{t}$. We can efficiently sample from $\mathbf{A}^{-1}(\mathbf{t})$ given a trapdoor for the matrix $\mathbf{A}$. To simplify the description in this overview, we use *curly underlines* to suppress small noise terms. Namely, we write $\underset{\sim}{\mathbf{s}^\intercal\mathbf{A}}$ to denote $\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal$ where $\mathbf{e}$ is a small error vector.

Distributed broadcast encryption. Next, we recall the syntax of a distributed broadcast encryption scheme [WQZDF10, BZ14]:

- **Setup:** In distributed broadcast encryption, there is an initial (trusted) setup algorithm that samples a set of public parameters pp. Similar to [WQZDF10, KMW23, GKPW24], we assume an a priori bound N on the maximum number of users, and allow the size of the public parameters to scale with N.
- **Key-generation:** In distributed broadcast encryption, each user has a distinct index $i \in [N]$. Using the public parameters pp, user i can generate a public/secret key-pair $(\mathsf{pk}_i, \mathsf{sk}_i)$. Typically, user i would post the public key pk_i to the public key directory. As noted in Section 1.2, the notion of flexible broadcast encryption [FWW23] eliminates the need for a user index (i.e., users simply generate a public/secret key-pair). The work of [GLWW23] show how to generically transform a distributed broadcast encryption into a flexible broadcast encryption scheme. In this work, we just focus on the simpler notion of distributed broadcast encryption.
- **Encryption:** The encryption algorithm takes the public parameters pp, a set of public keys $\{\mathsf{pk}_i\}_{i\in S}$, the message μ, and outputs the ciphertext ct.
- **Decryption:** The decryption algorithm takes a ciphertext, the public parameters pp, the associated set of public keys $\{\mathsf{pk}_i\}_{i\in S}$, the secret key sk_i for $i \in S$, and outputs the message.

The security requirement says that an encryption of μ to a set of public keys $\{\mathsf{pk}_i\}_{i\in S}$ should computationally hide μ from an adversary who only sees the public parameters pp and the public keys $\{\mathsf{pk}_i\}_{i\in S}$ of the users in the broadcast set. We say the scheme is selectively secure if the adversary has to declare the indices $S \subseteq [N]$ of the honest users at the beginning of the security game *before* it sees the public keys, and that it is adaptively secure if the adversary can choose the set S after seeing each user's public key (and selectively corrupting a subset of their keys). In this work, we are only able to prove selective security of our scheme; it is an interesting question to construct an adaptively secure distributed broadcast encryption scheme from lattice assumptions.[1]

[1]We are limited to selective security because our security proof relies on a "partitioning" argument where the reduction algorithm first programs the challenge set into the public parameters. This limitation is common to most lattice-based ABE and broadcast encryption schemes [GVW13, BGG+14, DKW21, WWW22, Wee22, HLL23, Wee24].

Starting point: a (centralized) broadcast encryption scheme. We begin by describing a simple (centralized) broadcast encryption scheme for N users. While previous lattice-based broadcast encryption schemes [BV22, Wee22, Wee24] start by constructing a ciphertext-policy ABE scheme with succinct ciphertexts, we take a more direct approach which notably does *not* rely on any of the homomorphic evaluation machinery typically seen in lattice-based ABE schemes. In turn, our approach more readily extends to support distributed key generation. The structure of our construction can be viewed as a lattice-based version of the pairing-based broadcast encryption scheme from [GW09, GKW18]. We describe our approach below:

- **Setup:** The master public key mpk for the broadcast encryption scheme is a tuple

$$\left(\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{W}_1, \ldots, \mathbf{W}_N, \mathbf{r}_1, \ldots, \mathbf{r}_N, \left\{\mathbf{A}^{-1}(\mathbf{W}_i \mathbf{r}_j)\right\}_{i \neq j}\right).$$

  Here, $\mathbf{A}, \mathbf{B}, \mathbf{W}_i \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\mathbf{p} \xleftarrow{R} \mathbb{Z}_q^n$, and $\mathbf{r}_i \leftarrow D_{\mathbb{Z},\sigma}^m$. The secret key for user $i \in [N]$ is

$$\mathsf{sk}_i = \mathbf{A}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i \mathbf{r}_i).$$

- **Encryption:** To encrypt a bit $\mu \in \{0,1\}$ to a set $S \subseteq [N]$, the encrypter samples an LWE secret $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$ and computes $\mathbf{W}_S = \sum_{j \in S} \mathbf{W}_j$. The ciphertext is

$$\mathsf{ct}_S = \left(\mathbf{s}^\intercal \mathbf{A}\,,\ \mathbf{s}^\intercal(\mathbf{B} + \mathbf{W}_S)\,,\ \mathbf{s}^\intercal \mathbf{p} + \mu \cdot \lfloor q/2 \rceil\right),$$

  where we write $\lfloor \cdot \rceil$ to denote the function that rounds to the nearest integer.

- **Decryption:** Decryption relies on the fact that when $i \in S$, we have

$$\begin{aligned}\mathbf{s}^\intercal \mathbf{A}\left(\mathsf{sk}_i + \sum_{j \in S \setminus \{i\}} \mathbf{A}^{-1}(\mathbf{W}_j \mathbf{r}_i)\right) &\approx \mathbf{s}^\intercal \mathbf{p} + \mathbf{s}^\intercal \mathbf{B}\mathbf{r}_i + \mathbf{s}^\intercal \mathbf{W}_i \mathbf{r}_i + \sum_{j \in S \setminus \{i\}} \mathbf{s}^\intercal \mathbf{W}_j \mathbf{r}_i \\ &= \mathbf{s}^\intercal \mathbf{p} + \mathbf{s}^\intercal \mathbf{B}\mathbf{r}_i + \mathbf{s}^\intercal \mathbf{W}_S \mathbf{r}_i,\end{aligned}$$

  where $\mathbf{A}^{-1}(\mathbf{W}_j \mathbf{r}_i)$ are the "cross-terms" from the master public key. To decrypt, user i then computes

$$\mathbf{s}^\intercal \mathbf{p} + \mu \cdot \lfloor q/2 \rceil + \mathbf{s}^\intercal(\mathbf{B} + \mathbf{W}_S)\mathbf{r}_i - \mathbf{s}^\intercal \mathbf{A}\left(\mathsf{sk}_i + \sum_{j \in S \setminus \{i\}} \mathbf{A}^{-1}(\mathbf{W}_j \mathbf{r}_i)\right) \approx \mu \cdot \lfloor q/2 \rceil,$$

  and rounds to recover μ.

We can prove selective security of this construction by relying on evasive LWE [Wee22, Tsa22].[2] The evasive LWE assumption essentially asserts that if $(\mathbf{s}^\intercal \mathbf{A}, \mathbf{s}^\intercal \mathbf{P})$ is pseudorandom, then $\mathbf{s}^\intercal \mathbf{A}$ is pseudorandom given $\mathbf{A}^{-1}(\mathbf{P})$. As noted above, in the selective security game, the adversary begins by declaring its challenge set $S^* \subseteq [N]$. It then receives the secret keys sk_i for all $i \notin S^*$ and its goal is to distinguish between encryptions of μ_0 and μ_1 to the set S^*. To prove selective security from evasive LWE, we leverage a partitioning argument

[2]Note that the security of our distributed broadcast encryption scheme will ultimately be based on the ℓ-succinct LWE assumption [Wee24], which is a falsifiable assumption that is implied by evasive LWE. However, we do not know how to prove security of this particular centralized broadcast encryption scheme from ℓ-succinct LWE. This is because our distributed broadcast encryption scheme will use a modified key-generation algorithm (described below).

where the reduction programs $\mathbf{B} := \mathbf{B}^* - \mathbf{W}_{S^*}$ where $\mathbf{B}^* \xleftarrow{R} \mathbb{Z}_q^{n\times m}$. Under evasive LWE, the claim now boils down to showing that

$$\mathbf{s}^\intercal\mathbf{A} \;,\; \mathbf{s}^\intercal\mathbf{B}^* \;,\; \mathbf{s}^\intercal\mathbf{p} \;,\; \left\{\mathbf{s}^\intercal\mathbf{W}_i\mathbf{r}_j\right\}_{i\neq j} \;,\; \left\{\mathbf{s}^\intercal(\mathbf{p} + (\mathbf{B}^* - \mathbf{W}_{S^*})\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i)\right\}_{i\notin S^*}$$

is pseudorandom. Since the details of this proof is immaterial to our subsequent construction (and analysis), we omit the formal details in this overview.[3]

Distributed key generation. To extend to distributed broadcast encryption, we partition the public parameters for the centralized broadcast encryption described above into two sets of components: one that is sampled by the initial (trusted) setup, and one that is sampled by each individual user:

- The components $(\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{r}_1, \ldots, \mathbf{r}_N)$ are part of the public parameters for the distributed broadcast encryption scheme.
- The matrices $\mathbf{W}_i$ as well as the cross terms $\mathbf{A}^{-1}(\mathbf{W}_i\mathbf{r}_j)$ for $j \neq i$ will be chosen by user i. Namely, the i^{th} user's public key is then $\mathsf{pk}_i = \left(\mathbf{W}_i, \left\{\mathbf{A}^{-1}(\mathbf{W}_i\mathbf{r}_j)\right\}_{j\neq i}\right)$. The decryption key for user i is still $\mathsf{sk}_i = \mathbf{A}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i)$. At this point, it is *unclear* how user i samples these components since it does *not* (and cannot) have a trapdoor for $\mathbf{A}$.

Observe that the public parameters pp together with any collection of public keys $\{\mathsf{pk}_i\}_{i\in S}$ now define a set of public parameters for the centralized broadcast encryption scheme (for $|S|$ users). Correctness now follows immediately. It suffices to build a mechanism for users to sample their public and secret keys *without* knowledge of a trapdoor for $\mathbf{A}$.

Sampling public keys. To complete the construction, we need a way for a user to sample a public key $\mathsf{pk}_i = \left(\mathbf{W}_i, \left\{\mathbf{A}^{-1}(\mathbf{W}_i\mathbf{r}_j)\right\}_{j\neq i}\right)$ together with a secret key $\mathsf{sk}_i = \mathbf{A}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i)$ *without* a trapdoor for $\mathbf{A}$. For simplicity, consider first the simpler goal of sampling a fresh $\mathbf{W}_i \in \mathbb{Z}_q^{n\times m}$ together with short vectors $\mathbf{y}_j \in \mathbb{Z}_q^m$ where $\mathbf{A}\mathbf{y}_j = \mathbf{W}_i\mathbf{r}_j$ for all $j \in [N]$. To facilitate this, we can publish a collection of random matrices $\mathbf{Z}_1, \ldots, \mathbf{Z}_k \in \mathbb{Z}_q^{n\times m}$ in the public parameters together with their preimages $\mathbf{A}^{-1}(\mathbf{Z}_i\mathbf{r}_j)$ for all $i \in [k]$ and $j \in [N]$. The user can now pick a (short) vector $\mathbf{d} \in \mathbb{Z}_q^k$ and define $\mathbf{W}_i := \sum_{\tau\in[k]} d_\tau\mathbf{Z}_\tau$. Moreover, if $\mathbf{d}$ is short, then $\sum_{\tau\in[k]} d_\tau\mathbf{A}^{-1}(\mathbf{Z}_\tau\mathbf{r}_j)$ is a short preimage of $\sum_{\tau\in[k]} d_\tau\mathbf{Z}_\tau\mathbf{r}_j = \mathbf{W}_i\mathbf{r}_j$ for all $i, j \in [N]$. In essence, the public parameters contain k public/secret key-pairs and the user samples their key by taking a random linear combination of the fixed keys in the public parameters. The hope then is that the user's public key $\mathbf{W}_i = \sum_{\tau\in[k]} d_\tau\mathbf{Z}_\tau$ and cross-terms $\sum_{\tau\in[k]} d_\tau\mathbf{A}^{-1}(\mathbf{Z}_\tau\mathbf{r}_j)$ hide the linear combination $\mathbf{d}$ the user used to generate their public/secret key-pair. While a Gaussian leftover hash lemma [AGHS13, AR16] can plausibly be used to show that the cross-terms are statistically close to $\mathbf{A}^{-1}(\sum_{\tau\in[k]} d_\tau\mathbf{Z}_\tau\mathbf{r}_j)$, we opt for a more direct approach inspired by recent constructions of functional commitments [WW23a, WW23b]. Namely, we publish a *full* trapdoor to facilitate direct sampling of the cross terms $\mathbf{A}^{-1}(\sum_{\tau\in[k]} d_\tau\mathbf{Z}_\tau\mathbf{r}_j)$ and the secret key. This approach is also more conducive to proving security from the ℓ-succinct LWE assumption.

[3] One approach is to first argue that $\mathbf{s}^\intercal\mathbf{W}_i\mathbf{r}_j$ is pseudorandom for all $i, j \in [N]$. Since $\mathbf{r}_j$ is short, we can use noise smudging to argue that $\mathbf{s}^\intercal\mathbf{W}_i\mathbf{r}_j \approx (\mathbf{s}^\intercal\mathbf{W}_i + \mathbf{e}^\intercal)\mathbf{r}_j$, for a small error vector $\mathbf{e}$. Then, by LWE (with secret $\mathbf{s}$), this is indistinguishable from $\mathbf{t}_i^\intercal\mathbf{r}_i$, where $\mathbf{t}_i \xleftarrow{R} \mathbb{Z}_q^m$. We can now appeal to LWE again (with secret $\mathbf{t}_i$) to argue that this is pseudorandom. Since $\mathbf{r}_j$ is short, this step would rely on the analysis from [BLMR13].

Publishing a trapdoor for a related matrix. Instead of publishing short preimages $\mathbf{A}^{-1}(\mathbf{Z}_i\mathbf{r}_j)$ in the public parameters, we give out a *full* trapdoor for a matrix related to $\mathbf{A}$ in the public parameters. In particular, we define the matrix

$$\mathbf{V} = \left[\begin{array}{ccc|ccc} \mathbf{A} & & & -\mathbf{Z}_1\mathbf{r}_1 & \cdots & -\mathbf{Z}_k\mathbf{r}_1 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \mathbf{A} & -\mathbf{Z}_1\mathbf{r}_N & \cdots & -\mathbf{Z}_k\mathbf{r}_N \end{array}\right] = \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_N) \end{array}\right] \in \mathbb{Z}_q^{nN \times (mN+k)}, \tag{1.1}$$

where $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \in \mathbb{Z}_q^{n \times mk}$. Suppose we sample

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{d} \end{bmatrix} \leftarrow \mathbf{V}^{-1}(\mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i)) \in \mathbb{Z}_q^{mN+k}, \tag{1.2}$$

where $\mathbf{u}_i \in \mathbb{Z}_q^N$ denotes the i^{th} canonical basis vector, and each $\mathbf{y}_j \in \mathbb{Z}_q^m$ and $\mathbf{d} \in \mathbb{Z}_q^k$. This means

$$\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_N) \end{array}\right] \cdot \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^n \\ \vdots \\ \mathbf{0}^n \\ \mathbf{p}+\mathbf{B}\mathbf{r}_i \\ \mathbf{0}^n \\ \vdots \\ \mathbf{0}^n \end{bmatrix} \in \mathbb{Z}_q^{nN}. \tag{1.3}$$

Next, by the mixed product rule for tensor (Kronecker) products (Eq. (2.1)), we can also write

$$\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d} = \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)(\mathbf{d} \otimes 1) = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)(1 \otimes \mathbf{r}_j) = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)\mathbf{r}_j.$$

Define $\mathbf{W}_i \coloneqq \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)$. Then, Eq. (1.3) says that for all $i \neq j$,

$$\forall j \neq i : \mathbf{A}\mathbf{y}_j - \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d} = \mathbf{0}^n \implies \mathbf{A}\mathbf{y}_j = \mathbf{W}_i\mathbf{r}_j$$

and

$$\mathbf{A}\mathbf{y}_i - \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)\mathbf{d} = \mathbf{p} + \mathbf{B}\mathbf{r}_i \implies \mathbf{A}\mathbf{y}_i = \mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i.$$

These are the *same* relations for the public parameters and the secret key as in the centralized broadcast encryption scheme. Moreover, when $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times m}$ and $m \geq O(n \log q)$, the distribution of $\mathbf{d}$ output by Eq. (1.2) is distributed according to a discrete Gaussian. This follows implicitly from the Gaussian preimage sampling algorithm from [GPV08]; we also refer to [WW23b, §2] for a formal proof. Correspondingly then, when $k \geq O(nm \log q)$, the distribution of $\mathbf{W}_i = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)$ is statistically close to uniform. Moreover the distribution of cross-terms $\mathbf{y}_j$ is distributed exactly according to $\mathbf{A}^{-1}(\mathbf{W}_i\mathbf{r}_j)$. As such, the public keys sampled using this procedure precisely coincide with the distribution in the original centralized broadcast encryption scheme. Putting all the pieces together, we now describe the full distributed broadcast encryption scheme:

- **Setup:** The public parameters pp consists of

  $$\text{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{r}_1, \ldots, \mathbf{r}_N, \mathbf{Z}, \text{td}_{\mathbf{V}}),$$

  where $\mathbf{A}, \mathbf{B} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times m}$, $\mathbf{p} \xleftarrow{\text{R}} \mathbb{Z}_q^n$, and $\mathbf{r}_1, \ldots, \mathbf{r}_N \leftarrow D_{\mathbb{Z},\sigma}^m$ exactly as in the centralized broadcast encryption scheme. The additional components $\mathbf{Z}$ and $\text{td}_{\mathbf{V}}$ are sampled as $\mathbf{Z} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times mk}$ and $\text{td}_{\mathbf{V}}$ is a (random) trapdoor for the matrix $\mathbf{V}$ in Eq. (1.1).

- **Key generation:** To generate a public/secret key pair for an index $i \in [N]$, the user uses the trapdoor $\mathsf{td}_{\mathbf{V}}$ to sample $(\mathbf{y}_1, \ldots, \mathbf{y}_N, \mathbf{d})$ according to Eq. (1.2). It computes $\mathbf{W}_i = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)$ and defines the public key to be $\mathsf{pk}_i = \left(\mathbf{W}_i, \{\mathbf{y}_j\}_{j \neq i}\right)$ and the secret key to be $\mathsf{sk}_i = \mathbf{y}_i$. As shown previously, for all $j \neq i$, it holds that $\mathbf{A}\mathbf{y}_j = \mathbf{W}_i \mathbf{r}_j$ and $\mathbf{A}\mathbf{y}_i = \mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i \mathbf{r}_i$.
- **Encryption and decryption:** These are the same as in the centralized broadcast encryption scheme. Specifically, the combination of the public parameters pp with the individual user public keys $\{\mathsf{pk}_i\}_{i \in [N]}$ can be viewed as a set of public parameters for the centralized broadcast encryption scheme. Since each user's secret key satisfies the same invariant as the centralized scheme, correctness follows as before.

We give the formal description in Section 3.1.

N-structured LWE. To prove security, we rely on the N-structured LWE assumption which asserts that

$$(\mathbf{A}, \mathbf{s}^\intercal \mathbf{A}, \mathbf{Z}, \mathbf{r}_1, \ldots, \mathbf{r}_N, \mathsf{td}_{\mathbf{V}}) \approx (\mathbf{A}, \mathbf{v}^\intercal, \mathbf{Z}, \mathbf{r}_1, \ldots, \mathbf{r}_N, \mathsf{td}_{\mathbf{V}}), \tag{1.4}$$

where $\mathbf{V}$ is the matrix in Eq. (1.1), $\mathsf{td}_{\mathbf{V}}$ is a random trapdoor for $\mathbf{V}$, and $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{v} \xleftarrow{R} \mathbb{Z}_q^m$, $\mathbf{Z} \xleftarrow{R} \mathbb{Z}_q^{n \times mk}$, and $\mathbf{r}_1, \ldots, \mathbf{r}_N \leftarrow D_{\mathbb{Z},\sigma}^m$. Later on, we will show that the N-structured LWE assumption follows from the ℓ-succinct LWE assumption recently introduced by Wee [Wee24]. We discuss both assumptions at the end of this section.

Proof strategy. We now provide a sketch of our security proof, and specifically, how the reduction algorithm simulates the key-generation queries. In the selective security game, the adversary begins by committing to the set of indices $S^* \subseteq [N]$ associated with the challenge ciphertext. The reduction algorithm obtains $(\mathbf{A}, \mathbf{v}^\intercal, \mathbf{Z}, \mathbf{r}_1, \ldots, \mathbf{r}_N, \mathsf{td}_{\mathbf{V}})$ from the ℓ-structured LWE challenger. It uses $\mathbf{A}, \mathbf{Z}, \mathbf{r}_1, \ldots, \mathbf{r}_N, \mathsf{td}_{\mathbf{V}}$ as the corresponding components of the public parameters for the distributed broadcast encryption scheme. The question is how the reduction algorithm simulates the public keys $\mathsf{pk}_i = (\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j \neq i})$ for the honest users and how it simulates the challenge ciphertext. Suppose for a moment that the reduction algorithm *knew* the $\mathbf{W}_i$ for each index $i \in S^*$ in the challenge set. Then, it would be able to compute $\mathbf{W}_{S^*} = \sum_{i \in S^*} \mathbf{W}_i$ and set $\mathbf{B} = \mathbf{A}\mathbf{H} - \mathbf{W}_{S^*}$, $\mathbf{p} = \mathbf{A}\mathbf{h}$ where $\mathbf{H} \xleftarrow{R} \{0,1\}^{m \times m}$ and $\mathbf{h} \xleftarrow{R} \{0,1\}^m$. In this case, the reduction could define the challenge ciphertext to be

$$\mathsf{ct}_{S^*} = \left(\mathbf{v}^\intercal, \mathbf{v}^\intercal \mathbf{H}, \mathbf{v}^\intercal \mathbf{h} + \mu \cdot \lfloor q/2 \rceil\right).$$

If $\mathbf{v}^\intercal = \mathbf{s}^\intercal \mathbf{A}$, then

$$\mathsf{ct}_{S^*} = \left(\mathbf{v}^\intercal, \mathbf{v}^\intercal \mathbf{H}, \mathbf{v}^\intercal \mathbf{h}\right) = \left(\mathbf{s}^\intercal \mathbf{A}, \mathbf{s}^\intercal \mathbf{A}\mathbf{H}, \mathbf{s}^\intercal \mathbf{A}\mathbf{h} + \mu \cdot \lfloor q/2 \rceil\right) = \left(\mathbf{s}^\intercal \mathbf{A}, \mathbf{s}^\intercal (\mathbf{B} + \mathbf{W}_{S^*}), \mathbf{s}^\intercal \mathbf{p} + \mu \cdot \lfloor q/2 \rceil\right),$$

which is distributed according to the real scheme. If $\mathbf{v}$ is a random vector, then by the leftover hash lemma, the challenge ciphertext is uniformly random and security holds.

The problem with this approach is that the reduction algorithm *cannot* choose $\mathbf{W}_i$ arbitrarily. Recall that $\mathbf{W}_i$ is a component of the public key, and in the real scheme, is derived by first sampling $(\mathbf{y}_{i,1}, \ldots, \mathbf{y}_{i,N}, \mathbf{d}_i)$ from $\mathbf{V}^{-1}(\mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i))$ according to Eq. (1.3) and then setting $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$. Here, we immediately run into a circularity issue. The reduction algorithm needs to know $\mathbf{W}_i$ in order to program the challenge set S^* into $\mathbf{B}$, but sampling $\mathbf{W}_i$ seemingly requires that $\mathbf{B}$ is already fixed!

Thus, the reduction algorithm needs an alternative method for simulating the honest users' public keys. The observation is simple: the public key pk_i for an index $i \in S^*$ only depends on $\mathbf{y}_{i,j}$ for $j \neq i$ and $\mathbf{d}$;

importantly, pk_i does not depend on the value of $\mathbf{y}_{i,i}$. Indeed, $\mathbf{y}_{i,i}$ is the secret key for user i which is not revealed to the adversary and also *cannot* be known to the reduction. Thus, in the reduction, instead of sampling $\mathbf{y}_{i,i}$ so that $\mathbf{A}\mathbf{y}_{i,i} = \mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i$ as in the real scheme, the reduction algorithm simply samples $\mathbf{y}_{i,i}$ so that $\mathbf{A}\mathbf{y}_{i,i} = \mathbf{W}_i\mathbf{r}_i$. In other words, the reduction algorithm samples $(\mathbf{y}_{i,1}, \ldots, \mathbf{y}_{i,N}, \mathbf{d}_i)$ from $\mathbf{V}^{-1}(\mathbf{0}^{nN})$. By the structure of $\mathbf{V}$ (see Eq. (1.1)), we can show that sampling from this distribution does *not* affect the marginal distributions of $\mathbf{y}_{i,j}$ for $j \neq i$ and $\mathbf{d}$. As such, this does not affect the adversary's view. With this modified sampling procedure, the reduction algorithm is able to sample the $\mathbf{W}_i$ components of each public key (independently of $\mathbf{B}$), and then program $\mathbf{W}_{S^*} = \sum_{i \in S^*} \mathbf{W}_i$ into the public parameters (as described above). We provide the full details in Section 3.1.

N-structured LWE and ℓ-succinct LWE. The above reduction relies on the N-structured LWE assumption (Eq. (1.4)) which essentially asserts hardness of LWE given a trapdoor for the related matrix $\mathbf{V}$ used in our construction. We can relate this assumption to the recently introduced ℓ-succinct LWE assumption [Wee24] which asserts that

$$(\mathbf{A}, \mathbf{s}^\mathsf{T}\mathbf{A}, \mathbf{U}, \mathsf{td}_\mathbf{V}) \approx (\mathbf{A}, \mathbf{v}^\mathsf{T}, \mathbf{U}, \mathsf{td}_\mathbf{V}), \tag{1.5}$$

where $\mathbf{A} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^n$, $\mathbf{v} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^m$, $\mathbf{U} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^{\ell n \times m}$, $\mathbf{V} = [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]$ and $\mathsf{td}_\mathbf{V}$ is a random trapdoor for the matrix $\mathbf{V}$. The ℓ-succinct LWE assumption is a falsifiable assumption and moreover, Wee showed that it is implied by the (public-coin) evasive LWE assumption [Wee24]. Wee also showed how to leverage ℓ-succinct LWE to construct an ABE scheme with succinct ciphertexts, which in particular, implies a (centralized) broadcast encryption scheme with short ciphertexts (and long public parameters). The analogous ℓ-succinct short integer solutions (SIS) assumption (i.e., SIS is hard with respect to $\mathbf{A}$ given a trapdoor for $\mathbf{V}$) has been used to construct succinct functional commitments [WW23a]. As shown in [Wee24], the ℓ-succinct SIS assumption is the least-structured or weakest among the multitude of structured lattice assumptions (e.g., $\mathsf{BASIS}_{\mathsf{struct}}$ [WW23b] or k-R-ISIS [ACL+22]) that have been introduced in recent years.

In Section 4, we show that if the ℓ-succinct LWE assumption holds with parameter $\ell \geq N \cdot O(\lambda \log N)$, then the N-structured LWE assumption also holds, provided that the width parameter k (i.e., the number of blocks in $\mathbf{Z}$) is at least $k \geq O(nm \log q)$. While it may appear that the N-structured LWE assumption gives out a trapdoor for a more structured matrix than the N-succinct LWE assumption, we show here that they are very similar. We illustrate this with a simple example. In the following description, we write $[\mathbf{I}_N \otimes \mathbf{A} \mid \mathbf{M}_{\mathbf{Z},\mathbf{R}}]$ (for $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_N]$) to denote the matrix $\mathbf{V}$ from Eq. (1.1) and $[\mathbf{I}_N \otimes \mathbf{A} \mid \mathbf{U}]$ to denote the matrix $\mathbf{V}$ from the ℓ-succinct LWE assumption (Eq. (1.5)). In particular,

$$\mathbf{M}_{\mathbf{Z},\mathbf{R}} \coloneqq \begin{bmatrix} -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ \vdots \\ -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_N) \end{bmatrix} \in \mathbb{Z}_q^{nN \times k} \quad \text{and} \quad \mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_\ell \end{bmatrix} \in \mathbb{Z}_q^{n\ell \times m}.$$

Suppose we sample $(\mathbf{y}_1, \ldots, \mathbf{y}_\ell, \mathbf{r})$ from $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]^{-1}(\mathbf{0}^{n\ell})$, where $\mathbf{y}_i \in \mathbb{Z}_q^m, \mathbf{r} \in \mathbb{Z}_q^m$. This is statistically indistinguishable from sampling

$$\mathbf{r} \leftarrow D_{\mathbb{Z},\sigma}^m \quad \text{and} \quad \forall i \in [\ell] : \mathbf{y}_i \leftarrow \mathbf{A}^{-1}(-\mathbf{U}_i\mathbf{r}). \tag{1.6}$$

On the other hand, suppose we sample $(\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N, \hat{\mathbf{d}})$ from $[\mathbf{I}_N \otimes \mathbf{A} \mid \mathbf{M}_{\mathbf{Z},\mathbf{R}}]^{-1}(\mathbf{0}^{nN})$, where $\hat{\mathbf{y}}_i \in \mathbb{Z}_q^m$ and $\hat{\mathbf{d}} \in \mathbb{Z}_q^k$. This is statistically indistinguishable from sampling

$$\hat{\mathbf{d}} \leftarrow D_{\mathbb{Z},\sigma}^k \quad \text{and} \quad \forall j \in [N] : \hat{\mathbf{y}}_i \leftarrow \mathbf{A}^{-1}(\mathbf{Z}(\hat{\mathbf{d}} \otimes \mathbf{I}_m)\mathbf{r}_i). \tag{1.7}$$

Since $\hat{\mathbf{d}} \leftarrow D_{\mathbb{Z},\sigma}^k$, when $k \geq O(nm \log q)$, the marginal distribution of $\mathbf{Z}(\hat{\mathbf{d}} \otimes \mathbf{I}_m)$ is statistically close to uniform. If we define $\mathbf{U} := \mathbf{Z}(\hat{\mathbf{d}} \otimes \mathbf{I}_m)$, then the distribution in Eq. (1.7) becomes

$$\hat{\mathbf{d}} \leftarrow D_{\mathbb{Z},\sigma}^k \quad \text{and} \quad \forall j \in [N] : \hat{\mathbf{y}}_i \leftarrow \mathbf{A}^{-1}(\mathbf{U}\mathbf{r}_i). \tag{1.8}$$

This is a similar "cross-term" structure as in Eq. (1.6), except with the roles of $\mathbf{U}$ and $\mathbf{r}$ interchanged (i.e., the same $\mathbf{r}$ is used for all $i \in [\ell]$ in Eq. (1.6) while the same $\mathbf{U}$ is used for all $i \in [N]$ in Eq. (1.8)). By "transposing" a collection of preimages sampled as in Eq. (1.6), we can transform them into a collection of preimages distributed as in Eq. (1.8). To simulate the $\mathbf{Z}$ and $\hat{\mathbf{d}}$ components that determine the $\mathbf{U}$ matrix in Eq. (1.8), we rely on preimage sampling techniques. We provide a formal reduction in Section 4 (Theorem 4.1).

1.2 Additional Related Work

Decentralized broadcast encryption. An alternative approach for solving the key-escrow problem in broadcast encryption is to rely on an *interactive* key-generation process. This is referred to as *decentralized broadcast encryption* [PPS12]. Namely, when a new user joins the system, the users in the system runs an MPC protocol with the existing users to obtain their secret key (and existing users obtain an updated key). In distributed broadcast encryption, key-generation is non-interactive and we do not require users to be cognizant of other users in the system.

Flexible broadcast encryption. In distributed broadcast encryption, each user's public key is actually associated with a slot index $i \in [N]$. Moreover, a user can only encrypt to a set of public keys if they occupy different slots. The work of [FWW23] introduced a stronger notion of *flexible* broadcast encryption where it is possible to encrypt to an *arbitrary* set of public keys without any slot restrictions. In the same work, the authors showed how to construct flexible broadcast encryption using witness encryption (together with a function-binding hash function). Recently, the work of [GLWW23] showed a generic compiler from distributed broadcast encryption to flexible broadcast encryption using combinatoric tools. The work of [GKPW24] also provides a direct construction of flexible broadcast encryption from pairings.

Registration-based cryptography. Distributed broadcast encryption falls into the more general umbrella of "registration-based cryptography" [GHMR18], which seeks to remove the trusted authority from advanced encryption schemes like identity-based encryption (IBE) [GHMR18, GHM+19, GV20, CES21, GKMR23, DKL+23, FKdP23], attribute-based encryption (ABE) [HLWW23, FWW23, ZZGQ23, GLWW24, AT24], functional encryption (FE) [FFM+23, DPY23], and traitor tracing [BLM+24]. Broadly speaking, the goal in each of these settings is to replace the trusted key-issuing authority with a public bulletin board where users can post their own public keys (that they themselves sample). Moreover a (transparent) key curator can then aggregate the individual public keys into a single short set of public parameters. The work of [FWW23] also shows how to compile any registered ABE scheme (that supports a single attribute and the always-accept policy) into a distributed broadcast encryption scheme (with *succinct* ciphertexts). Existing constructions of registered ABE either rely on indistinguishability obfuscation [HLWW23], witness encryption [FWW23], or pairing-based assumptions [HLWW23, ZZGQ23, GLWW24, AT24].

2 Preliminaries

Throughout this work, we write λ to denote the security parameter. For a positive integer $n \in \mathbb{N}$, we write $[n] := \{1, \ldots, n\}$. We write $\mathsf{poly}(\lambda)$ to denote a fixed polynomial in λ. We write $\mathsf{negl}(\lambda)$ to denote a

function that is negligible in λ: namely, $o(\lambda^{-c})$ for all $c \in \mathbb{N}$. We say an event occurs with overwhelming probability if the probability of its complement occurring is negligible. For functions $f = f(\lambda)$ and $g = g(\lambda)$, we write $f \geq O(g)$ to denote that there exists a fixed function $g' \in O(g)$ such that $f(\lambda) \geq g'(\lambda)$ for all $\lambda \in \mathbb{N}$. We say an algorithm is efficient if it runs in probabilistic polynomial time in the length of its input. For two ensembles of distributions $\mathcal{D}_1 = \{\mathcal{D}_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ and $\mathcal{D}_2 = \{\mathcal{D}_{2,\lambda}\}_{\lambda\in\mathbb{N}}$ indexed by a security parameter, we say they are computationally indistinguishable if no efficient algorithm can distinguish them except with $\mathsf{negl}(\lambda)$ probability. We say they are statistically indistinguishable if the statistical distance between them is $\mathsf{negl}(\lambda)$. We write $\mathcal{D}_1 \stackrel{c}{\approx} \mathcal{D}_2$ (resp., $\mathcal{D}_1 \stackrel{s}{\approx} \mathcal{D}_2$) if $\mathcal{D}_1$ and $\mathcal{D}_2$ are computationally (resp., statistically) indistinguishable. Throughout this work, we will use bold uppercase letters (e.g., $\mathbf{A}$, $\mathbf{B}$) to denote matrices and bold lowercase letters (e.g., $\mathbf{u}$, $\mathbf{v}$) to denote vectors. We use non-boldface letters (e.g., $v_1, \ldots, v_n$) to refer their components. For a dimension $n \in \mathbb{N}$, we write $\mathbf{I}_n \in \mathbb{Z}^{n\times n}$ to denote the identity matrix of dimension n. Throughout, we write $\|\cdot\|$ to denote the ℓ_∞ norm.

Tensor products. For matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{k\times \ell}$, we write $\mathbf{A} \otimes \mathbf{B} \in \mathbb{Z}_q^{nk\times m\ell}$ to denote their tensor (Kronecker) product. For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ where the products $\mathbf{AC}$ and $\mathbf{BD}$ are well-defined, then

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \tag{2.1}$$

We now recall a generalization of the leftover hash lemma along with a simple corollary that will be useful in our analysis.

Lemma 2.1 (Generalized Leftover Hash Lemma [ABB10, Lemma 13, adapted])**.** *Let n, m, q be integers such that $m \geq 2n \log q$ and $q > 2$ is prime. Then, for all fixed vectors $\mathbf{e} \in \mathbb{Z}_q^m$ and all $k = \mathsf{poly}(n)$, the statistical distance between the following distributions is $\mathsf{negl}(n)$:*

$$\left\{(\mathbf{A}, \mathbf{AR}, \mathbf{e}^\intercal\mathbf{R}) : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{R} \xleftarrow{R} \{0,1\}^{m\times k}\right\} \quad \text{and} \quad \left\{(\mathbf{A}, \mathbf{U}, \mathbf{e}^\intercal\mathbf{R}) : \begin{array}{c}\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{n\times k} \\ \mathbf{R} \xleftarrow{R} \{0,1\}^{m\times k}\end{array}\right\}.$$

Corollary 2.2 (Column Space of Random Matrix [GPV08, Lemma 5.1])**.** *Let n, m, q be lattice parameters where q is prime and $m \geq 2n \log q$. Then, for all but a $q^{-n} = \mathsf{negl}(n)$ fraction of matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$.*

Discrete Gaussians and gadget matrices. We write $D_{\mathbb{Z},\sigma}$ to denote the discrete Gaussian distribution over $\mathbb{Z}$ with width parameter $\sigma > 0$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a target vector $\mathbf{t} \in \mathbb{Z}_q^n$ in the column-space of $\mathbf{A}$, we write $\mathbf{A}_\sigma^{-1}(\mathbf{t})$ to denote a random variable $\mathbf{x} \leftarrow D_{\mathbb{Z},\sigma}^m$ conditioned on $\mathbf{Ax} = \mathbf{t} \bmod q$. We extend $\mathbf{A}_\sigma^{-1}$ to matrices by applying $\mathbf{A}_\sigma^{-1}$ to each column of the input. For positive integers $n, q \in \mathbb{N}$, let $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}^\intercal \in \mathbb{Z}_q^{n\times m'}$ be the gadget matrix [MP12] where $\mathbf{I}_n$ is the identity matrix of dimension n, $\mathbf{g}^\intercal = [1, 2, \ldots, 2^{\lfloor \log q \rfloor}]$, and $m' = n(\lfloor \log q \rfloor + 1)$. We also recall some basic properties of the discrete Gaussian distribution.

Lemma 2.3 (Gaussian Tail Bound [MP12, Lemma 2.6, adapted])**.** *Let n, m, q be lattice parameters where $m \geq 2n \log q$. Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$. Then, for all $\sigma > \log m$ and all vectors $\mathbf{t} \in \mathbb{Z}_q^n$ in the span of $\mathbf{A}$,*

$$\Pr[\|\mathbf{u}\| > \sqrt{m}\sigma : \mathbf{u} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{t})] \leq O(2^{-m}).$$

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

$$\Pr[|x| > \sqrt{\lambda}\sigma : x \leftarrow D_{\mathbb{Z},\sigma}] \leq 2^{-\lambda}.$$

Lemma 2.4 (Gaussian Samples [GPV08, adapted]). *Let n, m, q, σ be lattice parameters such that $\sigma \geq \log m$, $m \geq 2n \log q$, and q is prime. Then the statistical distance between the following distributions is at most* $\mathsf{negl}(n)$*:*

$$\left\{(\mathbf{A}, \mathbf{x}, \mathbf{Ax}) : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{x} \leftarrow D_{\mathbb{Z},\sigma}^m\right\} \quad \text{and} \quad \left\{(\mathbf{A}, \mathbf{x}, \mathbf{t}) : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{t} \xleftarrow{R} \mathbb{Z}_q^n, \mathbf{x} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{t})\right\}.$$

Basis extension and lattice trapdoors. We will also use the following lemma characterizing the distribution of $[\mathbf{A} \mid \mathbf{B}]^{-1}(\cdot)$. We give the statement from [WW23b], which follows immediately from earlier works on preimage sampling and basis delegation [GPV08, CHKP10, MP12]. Finally, we recall the notion of a gadget trapdoor [MP12].

Lemma 2.5 (Marginal of Gaussian Preimages [WW23b]). *Let n, m, q be lattice parameters where $m \geq 2n \log q$ and q is prime. Let $\mathbf{B} \in \mathbb{Z}_q^{n\ell\times k}$ where $\ell, k = \mathsf{poly}(n, \log q)$. Let $\mathbf{C} = [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{n\ell\times(m\ell+k)}$. Then for all target vectors $\mathbf{t} \in \mathbb{Z}_q^{n\ell}$ and all width parameters $s \geq \log(\ell m)$, the statistical distance between the following distributions is* $\mathsf{negl}(n)$*:*

$$\{\mathbf{v} : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{v} \leftarrow \mathbf{C}_s^{-1}(\mathbf{t})\} \quad \text{and} \quad \left\{\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix} : \mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{v}_2 \leftarrow D_{\mathbb{Z},s}^k, \mathbf{v}_1 \leftarrow (\mathbf{I}_\ell \otimes \mathbf{A})_s^{-1}(\mathbf{t} - \mathbf{B}\mathbf{v}_2)\right\}.$$

Lemma 2.6 (Gadget Trapdoor [Ajt96, GPV08, MP12]). *Let n, m, q be lattice parameters with $m \geq 3n \log q$. Then there exists efficient algorithms* $(\mathsf{TrapGen}, \mathsf{SamplePre})$ *with the following syntax:*

- $\mathsf{TrapGen}(1^n, q, m) \to (\mathbf{A}, \mathbf{R})$*: On input the lattice dimension n, the modulus q, and the number of samples m, the trapdoor-generation algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$ where $m' = n(\lfloor \log q \rfloor + 1)$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{t}, \sigma) \to \mathbf{x}$*: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$, a target vector $\mathbf{t} \in \mathbb{Z}_q^n$, and a Gaussian width parameter σ, the preimage-sampling algorithm outputs a vector $\mathbf{x} \in \mathbb{Z}_q^m$.*

Moreover, the above algorithms satisfy the following properties:

- **Trapdoor distribution:** *If $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\mathbf{A}' \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, then $\Delta(\mathbf{A}, \mathbf{A}') = \mathsf{negl}(n)$. Moreover, $\mathbf{AR} = \mathbf{G}_n \in \mathbb{Z}_q^{n\times m'}$ and $\|\mathbf{R}\| = 1$.*
- **Preimage sampling:** *For all matrices $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$, parameters $\sigma > 0$, and all target vectors $\mathbf{t} \in \mathbb{Z}_q^n$ in the column span of $\mathbf{A}$, the output $\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{t}, \sigma)$ satisfies $\mathbf{Ax} = \mathbf{t}$.*
- **Preimage distribution:** *Suppose $\mathbf{R}$ is a gadget trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ (i.e., $\mathbf{AR} = \mathbf{G}_n$). Then, for all $\sigma \geq m\|\mathbf{R}\| \log n$, and all target vectors $\mathbf{t} \in \mathbb{Z}_q^n$, the statistical distance between the following distributions is at most* $\mathsf{negl}(n)$*:*

$$\{\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{t}, \sigma)\} \quad \text{and} \quad \{\mathbf{x} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{t})\}.$$

Learning with errors and ℓ-succinct LWE. The learning with errors (LWE) assumption [Reg05] with parameters (n, m, q, σ) states that the distribution of $(\mathbf{A}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal)$ is computationally indistinguishable from $(\mathbf{A}, \mathbf{v}^\intercal)$ when $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, and $\mathbf{v} \xleftarrow{R} \mathbb{Z}_q^m$. Many recent works [ACL+22, WW23b, BCFL23, WW23a, CLM23, FMN23, Wee24] have introduced falsifiable variants of the LWE assumption (or the dual problem of short integer solutions (SIS)) which conjecture that the LWE (or SIS) problem with respect to $\mathbf{A}$ is hard even given a trapdoor for a matrix *related* to $\mathbf{A}$. In this work, we use the ℓ-succinct LWE assumption

introduced by Wee [Wee24], which asserts that LWE is hard with respect to $\mathbf{A}$ even given a trapdoor for the matrix $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]$ where $\mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{n\ell \times m}$. As discussed in [Wee24], the ℓ-succinct LWE assumption is weaker than many of the other recently-proposed structured lattice assumptions (specifically, the LWE analogs of k-R-ISIS [ACL+22, BCFL23] and $\mathsf{BASIS}_{\mathsf{struct}}$ [WW23a, FMN23]). It is also implied by assumptions like the evasive LWE assumption [Wee22, Tsa22]. We now give the formal statement of the assumption:

Assumption 2.7 (ℓ-Succinct LWE [Wee24])**.** Let λ be a security parameter and let $n = n(\lambda), m = m(\lambda), q = q(\lambda), \sigma = \sigma(\lambda)$ be lattice parameters. Let $s = s(\lambda)$ be a Gaussian width parameter and $\ell = \ell(\lambda)$ be a dimension. We say that the ℓ-succinct LWE assumption with parameters (n, m, q, σ, s) holds if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$:

$$\left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\intercal \mathbf{A} + \mathbf{e}^\intercal, \mathbf{U}, \mathbf{T}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{v}^\intercal, \mathbf{U}, \mathbf{T}) = 1]\right| = \mathsf{negl}(\lambda),$$

where $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, $\mathbf{v} \xleftarrow{R} \mathbb{Z}_q^m$, $\mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{n\ell\times m}$, and $\mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]_s^{-1}(\mathbf{G}_{n\ell})$.[4]

In other words, we require that LWE is hard with respect to $\mathbf{A}$ even given a fresh gadget trapdoor $\mathbf{T}$ for a related matrix $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]$.

2.1 Distributed Broadcast Encryption

We now define the notion of distributed broadcast encryption.

Definition 2.8 (Distributed Broadcast Encryption [BZ14, KMW23])**.** Let λ be the security parameter and N be the number of users. An N-user distributed broadcast encryption scheme is a tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

- $\mathsf{Setup}(1^\lambda, 1^N) \to \mathsf{pp}$: On input the security parameter λ and the number of users N, the setup algorithm outputs the public parameters pp.
- $\mathsf{KeyGen}(\mathsf{pp}, i) \to (\mathsf{pk}_i, \mathsf{sk}_i)$: On input the public parameters pp and an index $i \in [N]$, the key-generation algorithm outputs a public key and secret key $(\mathsf{pk}_i, \mathsf{sk}_i)$.
- $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) \to b$: On input the public parameters pp, an index $i \in [N]$, and a public key pk_i, the validity-checking algorithm outputs a bit $b \in \{0, 1\}$.
- $\mathsf{Enc}(\mathsf{pp}, \{(i, \mathsf{pk}_i)\}_{i\in S}, \mu) \to \mathsf{ct}$: On input the public parameters pp, a collection of public keys pk_i and a message $\mu \in \{0, 1\}$, the encryption algorithm outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{pp}, \{(i, \mathsf{pk}_i)\}_{i\in S}, \mathsf{ct}, (j, \mathsf{sk}_j)) \to \mu$: On input the public parameters pp, a collection of public keys pk_i, a ciphertext ct, and a secret key sk_j for an index j, the decryption algorithm outputs a message $\mu \in \{0, 1\}$.

We require that $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Enc}, \mathsf{Dec})$ satisfy the following properties:

- **Correctness:** For a security parameter $\lambda \in \mathbb{N}$, a bound N on the number of users, and an adversary $\mathcal{A}$, we define the correctness experiment as follows:
  - The challenger samples $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and gives pp to $\mathcal{A}$.

[4] Note that this distribution is only well defined when $\mathbf{G}_{n\ell}$ is in the image of $[\mathbf{I}_\ell \otimes \mathbf{A} \mid \mathbf{U}]$. Thus, when $\mathbf{G}_{n\ell}$ is not in the image, we set $\mathbf{T} = \perp$. Accordingly, taking $m \geq 2n \log q$ ensures that this event occurs with negligible probability (Corollary 2.2).

- The adversary specifies a target index $j \in [N]$. The challenger responds by computing $(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, j)$. It gives pk_j to the adversary $\mathcal{A}$.
- The adversary outputs a set $S \subseteq [N]$, a collection of public keys pk_i for $i \in S \setminus \{j\}$, and a message $\mu \in \{0,1\}$.
- The challenger checks that $j \in S$ and that $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) = 1$ for each $i \in S \setminus \{j\}$ and outputs $b = 1$ if not. Otherwise, the challenger computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \{(i, \mathsf{pk}_i)\}_{i \in S}, \mu)$ and $\mu' \leftarrow \mathsf{Dec}(\mathsf{pp}, \{(i, \mathsf{pk}_i)\}_{i \in S}, \mathsf{ct}, (j, \mathsf{sk}_j))$. It outputs $b = 1$ if $\mu = \mu'$ and $b = 0$ otherwise.

We say that the scheme is correct if for all $\lambda, N \in \mathbb{N}$ and all adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $\Pr[b = 1] \geq 1 - \mathsf{negl}(\lambda)$ in the correctness experiment.

- **Verifiable keys:** For all $\lambda, N \in \mathbb{N}$, and all indices $i \in [N]$, it holds that

$$\Pr\left[\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) = 1 : \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^N) \\ (\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, i) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

- **Selective security:** For a security parameter λ, a bound N on the number of users, and a bit $b \in \{0,1\}$, we define the selective security game between an adversary $\mathcal{A}$ and a challenger as follows:
  - On input the security parameter 1^λ and the number of users 1^N, the adversary outputs a challenge set $S^* \subseteq [N]$.
  - The challenger samples $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, i)$ for $i \in S^*$. It also computes $\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pp}, \{\mathsf{pk}_i\}_{i \in S^*}, b, S^*)$ and sends $\left(\mathsf{pp}, \{\mathsf{pk}_i\}_{i \in S^*}, \mathsf{ct}_b\right)$ to $\mathcal{A}$.
  - At the end of the game, algorithm $\mathcal{A}$ outputs $b' \in \{0,1\}$, which is the output of the experiment.

  We say the distributed broadcast encryption scheme is selectively secure if for all polynomials $N = N(\lambda)$, and all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]| = \mathsf{negl}(\lambda) \tag{2.2}$$

  in the selective security game. We say that the scheme is selectively secure for up to N users if Eq. (2.2) holds for the specific value of N.

- **Short ciphertexts:** There exists a fixed polynomial $\mathsf{poly}(\cdot)$ such that for all $\lambda, N \in \mathbb{N}$, all subsets $S \subseteq [N]$, all public parameters pp in the support of $\mathsf{Setup}(1^\lambda, 1^N)$, all key-pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ in the support of $\mathsf{KeyGen}(\mathsf{pp}, i)$ for $i \in S$, all messages $\mu \in \{0,1\}$, and all ciphertexts ct in the support of $\mathsf{Enc}(\mathsf{pp}, \{\mathsf{pk}_i\}_{i \in S}, \mu, S)$, it holds that $|\mathsf{ct}| \leq \mathsf{poly}(\lambda + \log N)$.

Remark 2.9 (Encrypting Long Messages)**.** Definition 2.8 considers the (simple) setting where the ciphertext encrypts a single bit. It is straightforward to support encrypting longer messages by composing with a symmetric encryption scheme. Namely, to encrypt a message $\mu \in \{0,1\}^m$, the encryption algorithm samples a symmetric key $k \in \{0,1\}^{\mathsf{poly}(\lambda)}$, encrypts the bits of k using the broadcast encryption scheme, and then encrypts μ using the symmetric key k. The size of the overall ciphertext is then $|\mu| + \mathsf{poly}(\lambda, \log N)$.

3 Distributed Broadcast Encryption from Lattices

In this section, we give our construction of a selectively-secure distributed broadcast encryption scheme. We begin by introducing an intermediate assumption called the ℓ-structured LWE assumption which we will use in our security analysis. Then, in Section 4, we show that our intermediate assumption follows from the ℓ'-succinct LWE assumption (Assumption 2.7) for $\ell' \geq \ell \cdot O(n \log q)$, where n, q are lattice parameters.

Assumption 3.1 (ℓ-Structured LWE)**.** Let λ be a security parameter and $n = n(\lambda), m = m(\lambda), q = q(\lambda), \sigma = \sigma(\lambda)$ be lattice parameters. Let $s = s(\lambda)$ be a Gaussian width parameter. Let $k = k(\lambda)$ and $\ell = \ell(\lambda)$ be dimension parameters. We say that the ℓ-structured LWE assumption with parameters (n, m, q, σ, s, k) holds if for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$:

$$\left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\intercal \mathbf{A} + \mathbf{e}^\intercal, \mathbf{Z}, \mathbf{R}, \mathbf{T}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}^\intercal, \mathbf{Z}, \mathbf{R}, \mathbf{T}) = 1]\right| = \mathsf{negl}(\lambda),$$

where $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, $\mathbf{u} \xleftarrow{R} \mathbb{Z}_q^m$, $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \xleftarrow{R} \mathbb{Z}_q^{n \times mk}$, $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \leftarrow D_{\mathbb{Z},s}^{m \times \ell}$, $\mathbf{T} \leftarrow (\mathbf{V}_{\ell,k})_s^{-1}(\mathbf{G}_{n\ell})$, and

$$\mathbf{V}_{\ell,k} = \left[\begin{array}{ccc|ccc} \mathbf{A} & & & -\mathbf{Z}_1\mathbf{r}_1 & \cdots & -\mathbf{Z}_k\mathbf{r}_1 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \mathbf{A} & -\mathbf{Z}_1\mathbf{r}_\ell & \cdots & -\mathbf{Z}_k\mathbf{r}_\ell \end{array}\right] = \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_\ell) \end{array}\right] \in \mathbb{Z}_q^{n\ell \times (m\ell + k)}. \tag{3.1}$$

Similar to ℓ-succinct LWE, we require that LWE is hard with respect to $\mathbf{A}$ even given a trapdoor $\mathbf{T}$ for a related matrix $\mathbf{V}_{\ell,k}$. While the right side of the matrix $\mathbf{V}_{\ell,k}$ appears significantly more structured than the random matrix $\mathbf{U}$ in the ℓ-succinct LWE assumption (Assumption 2.7), we show in Section 4 that this assumption is implied by the ℓ'-succinct LWE assumption (when $\ell' \geq \ell \cdot O(n \log q)$ and $k \geq 3nm \log q$).

Parameter setting. Similar to ℓ-succinct LWE, we only consider instantiations with $m \geq O(n \log q)$ so that the matrix $\mathbf{A}$ spans $\mathbb{Z}_q^n$ with overwhelming probability and the $\mathbf{Z}_i$ matrices have sufficient width. We additionally note that the ℓ-structured LWE assumption is false when k is too small. In this setting, the adversary can use the trapdoor to repeatedly sample $(\mathbf{V}_{\ell,k})_s^{-1}(\mathbf{0})$. By Lemma 2.5, these preimages include samples from $\mathbf{A}_s^{-1}(\mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)\mathbf{r}_i)$ where $\mathbf{d} \leftarrow D_{\mathbb{Z},s}^k$. When k is too small, collisions in the value of $\mathbf{d}$ will arise with noticeable probability. Such a collision immediately gives a short vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{0}$ (which immediately breaks LWE with respect to $\mathbf{A}$). Thus, we require $k = \omega(\log n)$ to ensure that collisions are unlikely to occur. In our setting, we will only consider $k \geq O(nm \log q)$. In this case, the marginal distribution of $\mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)$ is statistically close to uniform by Lemma 2.4. For this parameter reigme, we can in fact show that the ℓ-structured LWE assumptions holds under the ℓ'-succinct LWE for $\ell' \geq \ell \cdot O(n \log q)$. We provide this reduction in Section 4.

3.1 Distributed Broadcast Encryption from ℓ-Structured LWE

In this section, we describe our distributed broadcast encryption scheme from ℓ-structured LWE, where $\ell = N$ is the bound on the number of users in the system.

Construction 3.2 (Distributed Broadcast Encryption)**.** Let $\lambda \in \mathbb{N}$ be a security parameter, $N \in \mathbb{N}$ be the number of users, and $n = n(\lambda, N), m = m(\lambda, N), q = q(\lambda, N), \sigma = \sigma(\lambda, N)$ be lattice parameters. Let $s_0 = s_0(\lambda, N), s_1 = s_1(\lambda, N)$ be Gaussian width parameters, $k = k(\lambda, N)$ be a dimension, and $\beta = \beta(\lambda, N)$ be a norm bound. We construct our distributed broadcast encryption scheme $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{IsValid}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

- $\mathsf{Setup}(1^\lambda, 1^N)$: On input the security parameter λ and the bound on the number of users N, the setup algorithm proceeds as follows:

  1. Sample $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$, $\mathbf{B} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times m}$, $\mathbf{p} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$.
  2. For each $i \in [k]$, sample $\mathbf{Z}_i \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times m}$ and let $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \in \mathbb{Z}_q^{n\times mk}$. For each $i \in [N]$, sample $\mathbf{r}_i \leftarrow D_{\mathbb{Z},s_0}^m$.
  3. Sample $\mathbf{T}_\mathbf{V} \leftarrow \mathsf{SamplePre}\big(\mathbf{V}_{N,k}, \left[\begin{smallmatrix} \mathbf{I}_N \otimes \mathbf{T}_\mathbf{A} \\ \mathbf{0} \end{smallmatrix}\right], \mathbf{G}_{nN}, s_0\big)$, where

$$\mathbf{V}_{N,k} = \left[\begin{array}{ccc|ccc} \mathbf{A} & & & -\mathbf{Z}_1\mathbf{r}_1 & \cdots & -\mathbf{Z}_k\mathbf{r}_1 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & \mathbf{A} & -\mathbf{Z}_1\mathbf{r}_N & \cdots & -\mathbf{Z}_k\mathbf{r}_N \end{array}\right] = \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_N) \end{array}\right] \in \mathbb{Z}_q^{nN\times(mN+k)}. \tag{3.2}$$

  Output $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$.

- $\mathsf{KeyGen}(\mathsf{pp}, i)$: On input the public parameters $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$ and an index $i \in [N]$, the key-generation algorithm samples

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \\ \mathbf{d} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\mathbf{V}_{N,k}, \mathbf{T}_\mathbf{V}, \mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i), s_1), \tag{3.3}$$

  where $\mathbf{u}_i \in \{0,1\}^N$ is the the i^{th} standard basis vector, $\mathbf{y}_i \in \mathbb{Z}^m$ for each $i \in [N]$, and $\mathbf{d} \in \mathbb{Z}^k$. It sets $\mathbf{W} = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m) \in \mathbb{Z}_q^{n\times m}$ and outputs the public key $\mathsf{pk} = (\mathbf{W}, \{\mathbf{y}_j\}_{j\neq i})$ and the secret key $\mathsf{sk} = \mathbf{y}_i$.

- $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i)$: On input the public parameters $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$, an index $i \in [N]$, and a public key $\mathsf{pk}_i = (\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j\neq i})$, the validity-checking algorithm outputs 1 if the following holds:

$$\forall j \neq i : \mathbf{A}\mathbf{y}_{i,j} = \mathbf{W}_i\mathbf{r}_j \quad \text{and} \quad \|\mathbf{y}_{i,j}\| \leq \beta.$$

  Otherwise, the algorithm outputs 0.

- $\mathsf{Enc}(\mathsf{pp}, \{(j, \mathsf{pk}_j)\}_{j\in S}, \mu)$: On input the public parameters $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$, a collection of public keys $\mathsf{pk}_j = (\mathbf{W}_j, \{\mathbf{y}_{j,j'}\}_{j'\neq j})$ for each $j \in S$, and a message $\mu \in \{0,1\}$, the encryption algorithm samples $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, $\mathbf{H} \xleftarrow{\mathrm{R}} \{0,1\}^{m\times m}$, and $\mathbf{h} \xleftarrow{\mathrm{R}} \{0,1\}^m$. It computes $\mathbf{W}_S = \sum_{j\in S} \mathbf{W}_j$ and outputs

$$\mathsf{ct} = \big(\mathbf{s}^\mathsf{T}\mathbf{A} + \mathbf{e}^\mathsf{T},\ \mathbf{s}^\mathsf{T}(\mathbf{B} + \mathbf{W}_S) + \mathbf{e}^\mathsf{T}\mathbf{H},\ \mathbf{s}^\mathsf{T}\mathbf{p} + \mathbf{e}^\mathsf{T}\mathbf{h} + \mu \cdot \lfloor q/2 \rfloor\big).$$

- $\mathsf{Dec}(\mathsf{pp}, \{(j, \mathsf{pk}_j)\}_{j\in S}, \mathsf{ct}, (i, \mathsf{sk}_i))$: On input the public parameters $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$, a collection of public keys $\mathsf{pk}_j = (\mathbf{W}_j, \{\mathbf{y}_{j,j'}\}_{j'\neq j})$ for each $j \in S$, a ciphertext $\mathsf{ct} = (\mathbf{c}_1^\mathsf{T}, \mathbf{c}_2^\mathsf{T}, c_3)$, and a secret key $\mathsf{sk}_i = \mathbf{y}_{i,i} \in \mathbb{Z}_q^m$ for an index i, the decryption algorithm computes

$$z = c_3 + \mathbf{c}_2^\mathsf{T}\mathbf{r}_i - \mathbf{c}_1^\mathsf{T}\left(\mathbf{y}_{i,i} + \sum_{j\in S\setminus\{i\}} \mathbf{y}_{j,i}\right) \in \mathbb{Z}_q,$$

  and outputs $\lfloor z \rceil$ where $\lfloor z \rceil$ outputs 0 if $-q/4 \leq z < q/4$ and 1 otherwise.

Theorem 3.3 (Verifiable Keys). *Suppose q is prime, $n \geq \lambda$, $m \geq 2n\log q$, $s_0 \geq (mN+k)\log(nN)$, $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$, and $\beta \geq \sqrt{m}s_1$. Then, Construction 3.2 has verifiable keys.*

Proof. Let $\lambda, N \in \mathbb{N}$ and take any index $i \in [N]$. Let $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_{\mathbf{V}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$, and sample $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, i)$. Then, we can write

$$\mathsf{pk}_i = \left(\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j\neq i}\right) \quad \text{and} \quad \mathsf{sk}_i = \mathbf{y}_{i,i}.$$

We now show that $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) = 1$ with overwhelming probability:

- Since $s_0 \geq (mN+k)\log(nN)$, by Lemma 2.6, the distribution of $\mathbf{T}_{\mathbf{V}}$ is statistically close to $(\mathbf{V}_{N,k})_{s_0}^{-1}(\mathbf{G}_{nN})$. Since $m \geq 2n\log q$ and q is prime, by Lemmas 2.3 and 2.5, we have that $\|\mathbf{T}_{\mathbf{V}}\| \leq \sqrt{m}s_0$ with overwhelming probability.
- Since $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$, by Lemma 2.6, the distribution of $\mathbf{y}_{i,1}, \ldots, \mathbf{y}_{i,N}, \mathbf{d}_i$ output by Eq. (1.2) is statistically close to sampling from $(\mathbf{V}_{N,k})_{s_1}^{-1}(\mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i))$. By construction of $\mathbf{V}_{N,k}$ (see Eq. (3.2)) and using Eq. (2.1), this means that for all $j \neq i$

  $$\mathbf{0} = \mathbf{A}\mathbf{y}_{i,j} - \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d}_i = \mathbf{A}\mathbf{y}_{i,j} - \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)(\mathbf{d}_i \otimes 1) = \mathbf{A}\mathbf{y}_{i,j} - \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)\mathbf{r}_j.$$

  By definition of KeyGen, it sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$. Correspondingly, this means that

  $$\mathbf{A}\mathbf{y}_{i,j} = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)\mathbf{r}_j = \mathbf{W}_i\mathbf{r}_j.$$

- By Lemmas 2.3 and 2.5, $\|\mathbf{y}_{i,j}\| \leq \sqrt{m}s_1 \leq \beta$ with overwhelming probability.

Thus, $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) = 1$ holds with overwhelming probability. □

Theorem 3.4 (Correctness). *Suppose the modulus q is prime, $m \geq 2n\log q$, $s_0 \geq (mN+k)\log(nN)$, $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$, $\beta \geq \sqrt{m}s_1$, and $q \geq 4\sqrt{n}m\sigma(1 + N\beta + \sqrt{n}ms_0)$. Then, Construction 3.2 satisfies correctness.*

Proof. Let $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_{\mathbf{V}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$. Take any index $i \in [N]$, and let $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, i)$. Write $\mathsf{pk}_i = \left(\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j\neq i}\right)$ and $\mathsf{sk}_i = \mathbf{y}_{i,i}$. By the same analysis as in the proof of Theorem 3.3, we have $\|\mathbf{y}_{i,i}\| \leq \beta$ and $\mathbf{A}\mathbf{y}_{i,i} - \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)\mathbf{r}_i = \mathbf{p} + \mathbf{B}\mathbf{r}_i$. Since $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$, this means that

$$\mathbf{A}\mathbf{y}_{i,i} = \mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i \tag{3.4}$$

Take any set $S \subseteq [N]$ and any collection of public keys $\{\mathsf{pk}_j\}_{j\in S\setminus\{i\}}$ where pk_j satisfies $\mathsf{IsValid}(\mathsf{pp}, i, \mathsf{pk}_i) = 1$. This means that for all $j \in S \setminus \{i\}$,

$$\mathbf{A}\mathbf{y}_{j,i} = \mathbf{W}_j\mathbf{r}_i \quad \text{and} \quad \|\mathbf{y}_{j,i}\| \leq \beta. \tag{3.5}$$

Take any message $\mu \in \{0,1\}$ and let $\mathsf{ct} = (\mathbf{c}_1^\intercal, \mathbf{c}_2^\intercal, c_3) \leftarrow \mathsf{Enc}(\mathsf{pp}, \{\mathsf{pk}_i\}_{i\in S}, \mu, S)$. Let $\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in \mathbb{Z}_q^m, \mathbf{H} \in \{0,1\}^{m\times m}, \mathbf{h} \in \{0,1\}^m$ be the components sampled by encryption. Consider the output of the decryption algorithm $\mathsf{Dec}(\mathsf{pp}, \{(i, \mathsf{pk}_i)\}_{i\in S}, \mathsf{ct}, (j, \mathsf{sk}_j))$. First,

$$\mathbf{c}_1^\intercal\left(\mathbf{y}_{i,i} + \sum_{j\in S\setminus\{i\}} \mathbf{y}_{j,i}\right) = \mathbf{s}^\intercal\mathbf{A}\mathbf{y}_{i,i} + \sum_{j\in S\setminus\{i\}} \mathbf{s}^\intercal\mathbf{A}\mathbf{y}_{j,i} + \underbrace{\mathbf{e}^\intercal\mathbf{y}_{i,i} + \sum_{j\in S\setminus\{i\}} \mathbf{e}^\intercal\mathbf{y}_{j,i}}_{\tilde{e}_1}.$$

Combined with Eqs. (3.4) and (3.5), this becomes

$$\mathbf{c}_1^\intercal\left(\mathbf{y}_{i,i}+\sum_{j\in S\setminus\{i\}}\mathbf{y}_{j,i}\right)=\mathbf{s}^\intercal(\mathbf{p}+\mathbf{B}\mathbf{r}_i+\mathbf{W}_i\mathbf{r}_i)+\sum_{j\in S\setminus\{i\}}\mathbf{s}^\intercal\mathbf{W}_j\mathbf{r}_i+\tilde{e}_1=\mathbf{s}^\intercal(\mathbf{p}+\mathbf{B}\mathbf{r}_i+\mathbf{W}_S\mathbf{r}_i)+\tilde{e}_1,$$

using the fact that $\mathbf{W}_S=\sum_{j\in S}\mathbf{W}_j$ and $i\in S$. Next,

$$c_3+\mathbf{c}_2^\intercal\mathbf{r}_i=\mu\cdot\lfloor q/2\rfloor+\mathbf{s}^\intercal\mathbf{p}+\mathbf{s}^\intercal(\mathbf{B}+\mathbf{W}_S)\mathbf{r}_i+\underbrace{\mathbf{e}^\intercal\mathbf{h}+\mathbf{e}^\intercal\mathbf{H}\mathbf{r}_i}_{\tilde{e}_2}.$$

Putting everything together, we have

$$c_3+\mathbf{c}_2^\intercal\mathbf{r}_i-\mathbf{c}_1^\intercal\left(\mathbf{y}_{i,i}+\sum_{j\in S\setminus\{i\}}\mathbf{y}_{j,i}\right)=\mu\cdot\lfloor q/2\rfloor-\tilde{e}_1+\tilde{e}_2.$$

It suffices to show that $|\tilde{e}_1-\tilde{e}_2|<q/4$. We show this holds with overwhelming probability:

- Since Enc samples $\mathbf{e}\leftarrow D^m_{\mathbb{Z},\sigma}$, by Lemma 2.3, with overwhelming probability, $\|\mathbf{e}\|\le\sqrt{n}\sigma$.
- Since $\|\mathbf{y}_{j,i}\|\le\beta$ for all $j\in S$, it follows that $|\mathbf{e}^\intercal\mathbf{y}_{j,i}|\le\sqrt{n}m\beta\sigma$. Thus,

$$|\tilde{e}_1|\le\sum_{j\in S}|\mathbf{e}^\intercal\mathbf{y}_{j,i}|\le N\sqrt{n}m\beta\sigma.$$

- Next, $\mathbf{H}\in\{0,1\}^{m\times m}$ and $\mathbf{h}\in\{0,1\}^m$ so $|\mathbf{e}^\intercal\mathbf{h}|\le\sqrt{n}m\sigma$ and $\|\mathbf{e}^\intercal\mathbf{H}\|\le\sqrt{n}m\sigma$. Since $\mathbf{r}_i\leftarrow D^m_{\mathbb{Z},s_0}$, by Lemma 2.3, with overwhelming probability $\|\mathbf{r}_i\|\le\sqrt{n}s_0$. Then, $|\mathbf{e}^\intercal\mathbf{H}\mathbf{r}_i|\le nm^2\sigma s_0$. Thus,

$$|\tilde{e}_2|\le|\mathbf{e}^\intercal\mathbf{h}|+|\mathbf{e}^\intercal\mathbf{H}\mathbf{r}_i|\le\sqrt{n}m\sigma(1+\sqrt{n}ms_0).$$

Correctness holds as long as

$$q\ge4|\tilde{e}_1-\tilde{e}_2|\ge4\sqrt{n}m\sigma(1+N\beta+\sqrt{n}ms_0).$$ □

Theorem 3.5 (Selective Security). *Let λ be a security parameter and $N=N(\lambda)$ be any polynomial function. Suppose $n\ge\lambda$, $m\ge3n\log q$, $s_0\ge(mN+k)\log(nN)$ and $s_1\ge(mN+k)\sqrt{m}s_0\log(nN)$. Then, under the N-structured LWE assumption (Assumption 3.1) with parameters (n,m,q,σ,s_0,k), Construction 3.2 is selectively-secure for up to N users.*

Proof. Take any polynomial $N=N(\lambda)$ and any efficient adversary $\mathcal{A}$ for the selective security game. We start by defining a sequence of hybrid experiments:

- $\mathsf{Hyb}_0^{(b)}$: This is the selective security game with challenge bit $b\in\{0,1\}$. At the beginning of the game, the adversary $\mathcal{A}$ declares the set $S^*\subseteq[N]$. The challenger then samples $\mathsf{pp}\leftarrow\mathsf{Setup}(1^\lambda,1^N)$, $(\mathsf{pk}_i,\mathsf{sk}_i)\leftarrow\mathsf{KeyGen}(\mathsf{pp},i)$ for each $i\in S^*$, and $\mathsf{ct}\leftarrow\mathsf{Enc}(\mathsf{pp},\{\mathsf{pk}_i\}_{i\in S^*},b,S^*)$. The challenger gives $(\mathsf{pp},\{\mathsf{pk}_i\}_{i\in S^*},\mathsf{ct}_b)$ to the adversary $\mathcal{A}$. To recall, the challenger samples the elements as follows:
  - The challenger starts by sampling the components $(\mathbf{A},\mathbf{T}_\mathbf{A})\leftarrow\mathsf{TrapGen}(1^n,q,m)$, $\mathbf{B}\xleftarrow{\mathrm{R}}\mathbb{Z}_q^{n\times m}$, $\mathbf{p}\xleftarrow{\mathrm{R}}\mathbb{Z}_q^n$, $\mathbf{Z}_1,\ldots,\mathbf{Z}_k\xleftarrow{\mathrm{R}}\mathbb{Z}_q^{n\times m}$, and $\mathbf{r}_1,\ldots,\mathbf{r}_N\leftarrow D^m_{\mathbb{Z},s_0}$. It sets $\mathbf{Z}=[\mathbf{Z}_1\mid\cdots\mid\mathbf{Z}_k]\in\mathbb{Z}_q^{n\times mk}$ and $\mathbf{V}_{N,k}$ as in Eq. (3.2).

– Next, it samples a trapdoor $\mathbf{T}_{\mathbf{V}} \leftarrow \mathsf{SamplePre}(\mathbf{V}_{N,k}, \left[\begin{smallmatrix} \mathbf{I}_N \otimes \mathbf{T}_{\mathbf{A}} \\ \mathbf{0} \end{smallmatrix}\right], \mathbf{G}_{nN}, s_0)$. The challenger sets the public parameters to be

$$\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_{\mathbf{V}}).$$

– To generate the public key for $i \in S^*$, the challenger samples

$$\boldsymbol{\kappa}_i \leftarrow \mathsf{SamplePre}(\mathbf{V}_{N,k}, \mathbf{T}_{\mathbf{V}}, \mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i), s_1),$$

and then parses

$$\boldsymbol{\kappa}_i = \begin{bmatrix} \mathbf{y}_{i,1} \\ \vdots \\ \mathbf{y}_{i,N} \\ \mathbf{d}_i \end{bmatrix} \in \mathbb{Z}_q^{Nm+k}, \tag{3.6}$$

where $\mathbf{y}_{i,j} \in \mathbb{Z}_q^m$ and $\mathbf{d}_i \in \mathbb{Z}_q^k$. It sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$ and $\mathsf{pk}_i = \left(\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j\neq i}\right)$.

– Finally, to generate the challenge ciphertext, the challenger samples $\mathbf{s} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, $\mathbf{H} \xleftarrow{\mathrm{R}} \{0,1\}^{m\times m}$, and $\mathbf{h} \xleftarrow{\mathrm{R}} \{0,1\}^m$. It sets $\mathbf{W}_S = \sum_{j\in S^*} \mathbf{W}_j$ and constructs the challenge ciphertext as

$$\mathsf{ct}_b = (\mathbf{c}_1^\intercal, \mathbf{c}_2^\intercal, c_3) = \left(\mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal\,,\ \mathbf{s}^\intercal(\mathbf{B} + \mathbf{W}_S) + \mathbf{e}^\intercal\mathbf{H}\,,\ \mathbf{s}^\intercal\mathbf{p} + \mathbf{e}^\intercal\mathbf{h} + b\cdot\lfloor q/2\rfloor\right).$$

At the end of the experiment, algorithm $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$, which is the output of the experiment.

- $\mathsf{Hyb}_1^{(b)}$: Same as $\mathsf{Hyb}_0^{(b)}$, except the challenger samples $\mathbf{T}_{\mathbf{V}} \leftarrow (\mathbf{V}_{N,k})_{s_0}^{-1}(\mathbf{G}_{nN})$.
- $\mathsf{Hyb}_2^{(b)}$: Same as $\mathsf{Hyb}_1^{(b)}$, except for all $i \in S^*$, the challenger samples $\boldsymbol{\kappa}_i \leftarrow (\mathbf{V}_{N,k})_{s_1}^{-1}(\mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i))$.
- $\mathsf{Hyb}_3^{(b)}$: Same as $\mathsf{Hyb}_2^{(b)}$, except the challenger samples $\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times m}$.
- $\mathsf{Hyb}_4^{(b)}$: Same as $\mathsf{Hyb}_3^{(b)}$, except for all $i \in S^*$, the challenger first samples $\mathbf{d}_i \leftarrow D_{\mathbb{Z},s_1}^k$. Then, it sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$. Finally, it samples

$$\forall j \neq i : \mathbf{y}_{i,j} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{W}_i\mathbf{r}_j) \quad \text{and} \quad \mathbf{y}_{i,i} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i).$$

- $\mathsf{Hyb}_5^{(b)}$: Same as $\mathsf{Hyb}_4^{(b)}$, except for all $i \in S^*$, the challenger samples $\mathbf{y}_{i,i} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{W}_i\mathbf{r}_i)$.
- $\mathsf{Hyb}_6^{(b)}$: Same as $\mathsf{Hyb}_5^{(b)}$, except for all $i \in S^*$, the challenger samples $\boldsymbol{\kappa}_i \leftarrow (\mathbf{V}_{N,k})_{s_1}^{-1}(\mathbf{0}^{nN})$ and the components $\mathbf{y}_{i,j}$, $\mathbf{d}_i$ are again derived from $\boldsymbol{\kappa}_i$ according to Eq. (3.6).
- $\mathsf{Hyb}_7^{(b)}$: Same as $\mathsf{Hyb}_6^{(b)}$ except for all $i \in S^*$, the challenger samples $\boldsymbol{\kappa}_i \leftarrow \mathsf{SamplePre}(\mathbf{V}_{N,k}, \mathbf{T}_{\mathbf{V}}, \mathbf{0}^{nN}, s_1)$.
- $\mathsf{Hyb}_8^{(b)}$: Same as $\mathsf{Hyb}_7^{(b)}$, except the challenger sets $\mathbf{B} = \mathbf{B}^* - \mathbf{W}_{S^*}$, where $\mathbf{B}^* \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\times m}$.
- $\mathsf{Hyb}_9^{(b)}$: Same as $\mathsf{Hyb}_8^{(b)}$ except the challenger sets $\mathbf{B}^* = \mathbf{A}\mathbf{H}$ and $\mathbf{p} = \mathbf{A}\mathbf{h}$.
- $\mathsf{Hyb}_{10}^{(b)}$: Same as $\mathsf{Hyb}_9^{(b)}$ except the challenger samples $\mathbf{c}_1 \xleftarrow{\mathrm{R}} \mathbb{Z}_q^m$ and sets $\mathbf{c}_2^\intercal = \mathbf{c}_1^\intercal\mathbf{H}$ and $c_3 = \mathbf{c}_1^\intercal\mathbf{h} + b\cdot\lfloor q/2\rfloor$.

- $\mathsf{Hyb}_{11}^{(b)}$: Same as $\mathsf{Hyb}_{10}^{(b)}$ except the challenger samples $\mathbf{B}^* \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{p} \xleftarrow{R} \mathbb{Z}_q^n$, $\mathbf{c}_2 \xleftarrow{R} \mathbb{Z}_q^m$ and $c_3 \xleftarrow{R} \mathbb{Z}_q$.

We write $\mathsf{Hyb}_i^{(b)}(\mathcal{A})$ to denote the output distribution of an execution of $\mathsf{Hyb}_i^{(b)}$ with adversary $\mathcal{A}$. We now argue that each adjacent pair of distributions are indistinguishable.

Lemma 3.6. *Suppose $n \geq \lambda$, $m \geq 3n\log q$ and $s_0 \geq (mN+k)\log(nN)$. Then, $\mathsf{Hyb}_0^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_1^{(b)}(\mathcal{A})$.*

Proof. Since $m \geq 3n\log q$ and $s_0 \geq (mN+k)\log(nN)$, by Lemma 2.6, the distribution of $\mathbf{T}_\mathbf{V}$ in $\mathsf{Hyb}_0^{(b)}$ is statistically indistinguishable from sampling $\mathbf{T}_\mathbf{V} \leftarrow (\mathbf{V}_{N,k})_{s_0}^{-1}(\mathbf{G}_{nN})$. □

Lemma 3.7. *Suppose $n \geq \lambda$, $m \geq 2n\log q$ and $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$. Then, $\mathsf{Hyb}_1^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_2^{(b)}(\mathcal{A})$.*

Proof. Since $m \geq 2n\log q$ and q is prime, by Lemmas 2.3 and 2.5, we have that $\|\mathbf{T}_\mathbf{V}\| \leq \sqrt{m}s_0$ with overwhelming probability. Since $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$, by Lemma 2.6, the distribution of $\boldsymbol{\kappa}_i$ in $\mathsf{Hyb}_1^{(b)}$ is statistically close to sampling from $(\mathbf{V}_{N,k})_{s_1}^{-1}(\mathbf{u}_i \otimes (\mathbf{p} + \mathbf{B}\mathbf{r}_i))$. Since $N = \mathsf{poly}(\lambda)$, the claim now follows by a hybrid argument. □

Lemma 3.8. *Suppose $n \geq \lambda$ and $m \geq 3n\log q$. Then, $\mathsf{Hyb}_2^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_3^{(b)}(\mathcal{A})$.*

Proof. Follows immediately by Lemma 2.6. □

Lemma 3.9. *Suppose $n \geq \lambda$, $m \geq 2n\log q$, and $s_1 \geq \log(mN)$. Then, $\mathsf{Hyb}_3^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_4^{(b)}(\mathcal{A})$.*

Proof. By Lemma 2.5, for each $i \in S^*$, the distribution of $\{\mathbf{y}_{i,j}\}_{j\in[N]}$ and $\mathbf{d}_i$ in $\mathsf{Hyb}_3^{(b)}$ is statistically close to the distribution

$$\mathbf{d}_i \leftarrow D_{\mathbb{Z},s_1}^k \quad \text{and} \quad \forall j \neq i : \mathbf{y}_{i,j} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d}_i) \quad \text{and} \quad \mathbf{y}_{i,i} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)\mathbf{d}_i).$$

In $\mathsf{Hyb}_3^{(b)}$, the challenger then sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$. In this case, by Eq. (2.1),

$$\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d}_i = \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)(\mathbf{d}_i \otimes 1) = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)\mathbf{r}_j = \mathbf{W}_i\mathbf{r}_j.$$

The challenger's sampling procedure in $\mathsf{Hyb}_3^{(b)}$ is thus equivalent to first sampling $\mathbf{d}_i \leftarrow D_{\mathbb{Z},s_1}^k$, then setting $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$, and finally sampling

$$\forall j \neq i : \mathbf{y}_{i,j} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{W}_i\mathbf{r}_j) \quad \text{and} \quad \mathbf{y}_{i,i} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{p} + \mathbf{B}\mathbf{r}_i + \mathbf{W}_i\mathbf{r}_i).$$

This is the sampling procedure in $\mathsf{Hyb}_4^{(b)}$. Since $N = \mathsf{poly}(\lambda)$, the claim now follows by a hybrid argument over each $i \in S^*$. □

Lemma 3.10. *The distributions $\mathsf{Hyb}_4^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_5^{(b)}(\mathcal{A})$ are identically distributed.*

Proof. The adversary's view in the two experiments is *independent* of $\mathbf{y}_{i,i}$ for all $i \in S^*$, so these two distributions are identical. □

Lemma 3.11. *Suppose $n \geq \lambda$ and $m \geq 2n\log q$, and $s_1 \geq \log(mN)$. Then, $\mathsf{Hyb}_5^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_6^{(b)}(\mathcal{A})$.*

Proof. This follows by a similar argument as in the proof of Lemma 3.9. Specifically, by Lemma 2.5, the distribution of $\{\mathbf{y}_{i,j}\}_{j\in[N]}$ and $\mathbf{d}_i$ in $\mathsf{Hyb}_6^{(b)}$ is statistically close to the distribution

$$\mathbf{d}_i \leftarrow D^k_{\mathbb{Z},s_1} \quad \text{and} \quad \forall j \in [N] : \mathbf{y}_{i,j} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d}_i).$$

Then, the challenger sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d}_i \otimes \mathbf{I}_m)$. As in the proof of Lemma 3.9, we can write $\mathbf{W}_i\mathbf{r}_j = \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_j)\mathbf{d}_i$. Thus, the challenger is sampling $\mathbf{y}_{i,j} \leftarrow \mathbf{A}_{s_1}^{-1}(\mathbf{W}_i\mathbf{r}_j)$ for all $j \in [N]$. This is the distribution in $\mathsf{Hyb}_5^{(b)}$. □

Lemma 3.12. *Suppose $n \geq \lambda$, $m \geq 2n\log q$ and $s_1 \geq (mN+k)\sqrt{m}s_0\log(nN)$. Then, $\mathsf{Hyb}_6^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_7^{(b)}(\mathcal{A})$.*

Proof. Follows by the same argument as in the proof of Lemma 3.7. □

Lemma 3.13. *The distributions $\mathsf{Hyb}_7^{(b)}(\mathcal{A})$ and $\mathsf{Hyb}_8^{(b)}(\mathcal{A})$ are identically distributed.*

Proof. In $\mathsf{Hyb}_7^{(b)}$ and $\mathsf{Hyb}_8^{(b)}$, none of the $\boldsymbol{\kappa}_i$ depend on $\mathbf{B}$. As such, the distribution of $\mathbf{B}$ is uniform and *independent* of $\mathbf{W}_{S^*}$ in both experiments. As such, these two distributions are identical. □

Lemma 3.14. *Suppose $n \geq \lambda$, $m \geq 2n\log q$ and $q > 2$ is prime. Then, $\mathsf{Hyb}_8^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_9^{(b)}(\mathcal{A})$.*

Proof. By Lemma 2.1, for all $\mathbf{e} \in \mathbb{Z}_q^m$, the following two distributions are statistically indistinguishable:

$$\left\{ (\mathbf{A}, \mathbf{AH}, \mathbf{Ah}, \mathbf{e}^\intercal\mathbf{H}, \mathbf{e}^\intercal\mathbf{h}) : \begin{array}{c} \mathbf{A} \stackrel{R}{\leftarrow} \mathbb{Z}_q^{n\times m} \\ \mathbf{H} \stackrel{R}{\leftarrow} \{0,1\}^{m\times m} \\ \mathbf{h} \stackrel{R}{\leftarrow} \{0,1\}^m \end{array} \right\} \quad \text{and} \quad \left\{ (\mathbf{A}, \mathbf{B}^*, \mathbf{p}, \mathbf{e}^\intercal\mathbf{H}, \mathbf{e}^\intercal\mathbf{h}) : \begin{array}{c} \mathbf{A} \stackrel{R}{\leftarrow} \mathbb{Z}_q^{n\times m}, \\ \mathbf{B}^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^{n\times m}, \mathbf{p} \stackrel{R}{\leftarrow} \mathbb{Z}_q^n \\ \mathbf{H} \stackrel{R}{\leftarrow} \{0,1\}^{m\times m}, \mathbf{h} \stackrel{R}{\leftarrow} \{0,1\}^m \end{array} \right\}.$$

The left distribution corresponds to $\mathsf{Hyb}_9^{(b)}$ while the right distribution corresponds to $\mathsf{Hyb}_8^{(b)}$. □

Lemma 3.15. *Suppose the N-structured LWE (Assumption 3.1) holds with parameters $(n, m, q, \sigma, s_0, k)$. Then, $\mathsf{Hyb}_9^{(b)}(\mathcal{A}) \stackrel{c}{\approx} \mathsf{Hyb}_{10}^{(b)}(\mathcal{A})$.*

Proof. Suppose there exists a bit $b \in \{0,1\}$ and an efficient adversary $\mathcal{A}$ that can distinguish between $\mathsf{Hyb}_9^{(b)}$ and $\mathsf{Hyb}_{10}^{(b)}$ with non-negligible advantage $\varepsilon > 0$. We use algorithm $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ that breaks the N-structured LWE assumption with parameters $(n, m, q, \sigma, s_0, k)$:

1. At the start of the game, algorithm $\mathcal{B}$ receives an N-structured LWE challenge $(\mathbf{A}, \mathbf{c}_1^\intercal, \mathbf{Z}, \mathbf{R}, \mathbf{T}_\mathbf{V})$ from its challenger. Let $\mathbf{V}_{N,k}$ be the matrix from Eq. (3.2) formed from the components $\mathbf{Z}$ and $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_N]$.
2. Algorithm $\mathcal{B}$ samples $\mathbf{H} \stackrel{R}{\leftarrow} \{0,1\}^{m\times m}$ and $\mathbf{h} \stackrel{R}{\leftarrow} \{0,1\}^m$, then sets $\mathbf{B}^* = \mathbf{AH}, \mathbf{p} = \mathbf{Ah}, \mathbf{c}_2^\intercal = \mathbf{c}_1^\intercal\mathbf{H}$ and $c_3 = \mathbf{c}_1^\intercal\mathbf{h} + b \cdot \lfloor q/2 \rfloor$.
3. Algorithm $\mathcal{B}$ starts running $\mathcal{A}$ and receives a challenge set $S^* \subseteq [N]$ from algorithm $\mathcal{A}$. For all $i \in S^*$, algorithm $\mathcal{B}$ samples $\boldsymbol{\kappa}_i \leftarrow \mathsf{SamplePre}(\mathbf{V}_{N,k}, \mathbf{T}_\mathbf{V}, \mathbf{0}^{nN}, s_1)$ and sets $\mathbf{W}_i = \mathbf{Z}(\mathbf{d} \otimes \mathbf{I}_m)$, where $\mathbf{y}_{i,j}$ and $\mathbf{d}_i$ are derived from $\boldsymbol{\kappa}_i$ according to Eq. (3.6).
4. Algorithm $\mathcal{B}$ sets $\mathbf{B} = \mathbf{B}^* - \mathbf{W}_{S^*}$, $\mathsf{pp} = (\mathbf{A}, \mathbf{B}, \mathbf{p}, \mathbf{Z}, \{\mathbf{r}_i\}_{i\in[N]}, \mathbf{T}_\mathbf{V})$, and $\mathsf{pk}_i = (\mathbf{W}_i, \{\mathbf{y}_{i,j}\}_{j\neq i})$ for each $i \in S^*$. It sets $\mathsf{ct}_b = (\mathbf{c}_1^\intercal, \mathbf{c}_2^\intercal, c_3)$. It gives $(\mathsf{pp}, \{\mathsf{pk}_i\}_{i\in S^*}, \mathsf{ct}_b)$ to $\mathcal{A}$ and outputs whatever algorithm $\mathcal{A}$ outputs.

We first show that $\mathcal{B}$ correctly simulates an execution of $\mathsf{Hyb}_9^{(b)}$ and $\mathsf{Hyb}_{10}^{(b)}$ for $\mathcal{A}$.

- The N-structured LWE challenger samples $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{Z} \xleftarrow{R} \mathbb{Z}_q^{n\times mk}$, and $\mathbf{R} \leftarrow D_{\mathbb{Z},s_0}^{m\times \ell}$. Moreover, the challenger samples $\mathbf{T}_V \leftarrow (\mathbf{V}_{N,k})_{s_0}^{-1}(\mathbf{G}_{nN})$, which coincides with the distribution of the public parameters in $\mathsf{Hyb}_9^{(b)}$ and $\mathsf{Hyb}_{10}^{(b)}$. Next, algorithm $\mathcal{B}$ sets $\mathbf{B} = \mathbf{AH}$ and $\mathbf{p} = \mathbf{Ah}$, so we conclude that the public parameters pp are perfectly simulated.
- Next, algorithm $\mathcal{B}$ samples $\boldsymbol{\kappa}_i$ using the same procedure as in $\mathsf{Hyb}_9^{(b)}$ and $\mathsf{Hyb}_{10}^{(b)}$, so the public keys are perfectly simulated.
- Consider the distribution of the challenge ciphertext:
  - If $\mathbf{c}_1^\intercal = \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal$ where $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow D_{\mathbb{Z},\sigma}^m$, then

$$\mathbf{c}_2^\intercal = \mathbf{c}_1^\intercal\mathbf{H} = \mathbf{s}^\intercal\mathbf{AH} + \mathbf{e}^\intercal\mathbf{H} = \mathbf{s}^\intercal\mathbf{B}^* + \mathbf{e}^\intercal\mathbf{H} = \mathbf{s}^\intercal(\mathbf{B} + \mathbf{W}_{S^*}) + \mathbf{e}^\intercal\mathbf{H}$$
$$c_3 = \mathbf{c}_1^\intercal\mathbf{h} + b\cdot\lfloor q/2\rfloor = \mathbf{s}^\intercal\mathbf{Ah} + \mathbf{e}^\intercal\mathbf{p} + b\cdot\lfloor q/2\rfloor = \mathbf{s}^\intercal\mathbf{p} + \mathbf{e}^\intercal\mathbf{h} + b\cdot\lfloor q/2\rfloor,$$

    which is the distribution of the challenge ciphertext in $\mathsf{Hyb}_9^{(b)}$.
  - Conversely, if $\mathbf{c}_1 \xleftarrow{R} \mathbb{Z}_q^m$. Then, algorithm $\mathcal{B}$ perfectly simulates an execution of $\mathsf{Hyb}_{10}^{(b)}$.

We conclude that algorithm $\mathcal{B}$ breaks the N-structured LWE problem with the same advantage ε. □

Lemma 3.16. *If $n \geq \lambda$, $m \geq 2(n+1)\log q$, and $q > 2$ is a prime, then $\mathsf{Hyb}_{10}^{(b)}(\mathcal{A}) \stackrel{s}{\approx} \mathsf{Hyb}_{11}^{(b)}(\mathcal{A})$*

Proof. This follows from Lemma 2.1 applied to the matrix $\left[\begin{smallmatrix}\mathbf{A}\\ \mathbf{c}_1^\intercal\end{smallmatrix}\right] \in \mathbb{Z}_q^{(n+1)\times m}$. □

Lemma 3.17. *The experiments $\mathsf{Hyb}_{10}^{(0)}(\mathcal{A})$ and $\mathsf{Hyb}_{11}^{(1)}(\mathcal{A})$ are identically distributed.*

Proof. By construction, the challenger's behavior in $\mathsf{Hyb}_{11}^{(b)}$ is *independent* of the challenge bit $b \in \{0,1\}$, so the adversary's view in the two distributions is identical. □

Combining Lemmas 3.6 to 3.17, selective security follows by a hybrid argument. □

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 3.2 to satisfy Theorems 3.3 to 3.5:

- We set the lattice dimension $n = \lambda$ and $m = O(n\log q)$.
- We set the noise parameter $\sigma = \mathsf{poly}(n)$ (such that the LWE assumption with parameters (n, m, q, σ) holds). We set the dimension to be $k = O(nm\log q)$.
- We set $s_0 = (mN+k)\log(nN)$ and $s_1 = (mN+k)\sqrt{m}s_0\log(nN) = (mN+k)^2\sqrt{m}\log^2(nN)$.
- Finally, we set the norm bound $\beta = \sqrt{m}s_1 = (mN+k)^2 m\log^2(nN)$ and the modulus q such that

$$q \geq 4\sqrt{n}m\sigma(1 + N\beta + \sqrt{n}ms_0) = N^3\cdot\mathsf{poly}(\lambda, \log N).$$

  In this case, $\log q = O(\log N + \log\lambda)$.

With this setting of parameters, we obtain a distributed broadcast encryption scheme with the following parameter sizes. Without loss of generality, we assume that $N \geq \lambda$.

- **Public parameter size:** The public parameters pp have size $|\mathsf{pp}| = N^2 \cdot \mathsf{poly}(\lambda, \log N)$.
- **Public key size:** Each user's public key pk consists of a matrix $\mathbf{W} \in \mathbb{Z}_q^{n \times m}$ and $N-1$ cross-terms $\mathbf{y}_j \in \mathbb{Z}_q^m$, so $|\mathsf{pk}| \leq (n+N)m \log q = O(N\lambda \log^2 N)$.
- **Secret key size:** The secret key for user $i \in [N]$ consists of a vector $\mathbf{y}_i \in \mathbb{Z}_q^m$, so $|\mathsf{sk}_i| = O(m \log q) = O(\lambda \log^2 N)$.
- **Ciphertext size:** The ciphertext for any set $S \subseteq [N]$ and message $\mu \in \{0,1\}$ consists of of $2m+1$ elements of $\mathbb{Z}_q$, so $|\mathsf{ct}| = O(\lambda \log^2 N)$.

Combined with the reduction from ℓ-structured LWE to ℓ'-succinct LWE (for $\ell' \geq \ell \cdot O(n \log q)$) from Section 4 (Theorem 4.1), we obtain the following corollary:

Corollary 3.18 (Distributed Broadcast Encryption from ℓ-succinct LWE)**.** *Let λ be a security parameter and $N = N(\lambda)$ be any polynomial. Let $\ell \geq N \cdot O(\lambda \log N)$. Under the ℓ-succinct LWE assumption for $\ell = N \cdot O(\lambda \log N)$ (and a polynomial modulus-to-noise ratio), there exists a selectively-secure distributed broadcast encryption scheme. Both the size of the ciphertext and a user's secret key is $O(\lambda \log^2 N)$, the size of a user's public key is $O(N\lambda \log^2 N)$, and the size of the public parameters is $N^2 \cdot \mathsf{poly}(\lambda, \log N)$.*

Remark 3.19 (Precomputing Encryption and Decryption Keys)**.** Similar to the pairing-based constructions of distributed broadcast encryption [KMW23, GLWW23, GKPW24], we can improve the efficiency of the encryption and decryption algorithms when the broadcast set S is known in advance. Specifically, we can view the matrix $\mathbf{W}_S = \sum_{i \in S} \mathbf{W}_i$ as the public key for encrypting to the set S; given $\mathbf{W}_S$, encrypting a message to the set S just requires time $\mathsf{poly}(\lambda, \log N)$. Similarly, if user j knows the broadcast set S in advance, she can pre-compute her decryption component $\mathbf{y}_{j,S} := \sum_{i \in S \setminus \{j\}} \mathbf{y}_{i,j}$. Given $\mathbf{y}_{j,S}$, decryption now runs in time $\mathsf{poly}(\lambda, \log N)$. Precomputation is useful in settings where users frequently send or receive broadcasts to the *same* set S.

4 Relating ℓ-Structured LWE and ℓ-Succinct LWE

In this section, we formally show that the ℓ-structured LWE assumption (Assumption 3.1) used in Section 3.1 follows under the ℓ'-succinct LWE assumption (Assumption 2.7) when $\ell' \geq \ell \cdot O(n \log q)$, where n is the lattice dimension and q is the modulus. Essentially, our proof shows how to build a trapdoor for the ℓ-structured LWE assumption using a trapdoor for the ℓ'-succinct LWE assumption. We refer to Section 1.1 for a high-level overview of our proof strategy.

Theorem 4.1 (ℓ'-succinct LWE implies ℓ-structured LWE)**.** *Let λ be a security parameter and $n = n(\lambda), m = m(\lambda), q = q(\lambda), \sigma = \sigma(\lambda)$ be lattice parameters. Let $s = s(\lambda), s' = s'(\lambda)$ be Gaussian width parameters, and $\ell = \ell(\lambda), k = k(\lambda)$ be polynomially-bounded dimensions. Suppose q is prime and the following conditions hold:*

- *$n \geq \lambda$, $m \geq 2n \log q$, $k \geq 3nm \log q$, $q \leq 2^n$;*
- *$s' \geq \log m$, $s \geq \max\left\{m^{3/2}(\ell'+1)s' \log(n\ell'), k \log(nm)\right\}$.*

Let $\ell' = \ell n(\lfloor \log q \rfloor + 1)$. Then, the ℓ'-succinct LWE assumption with parameters (n, m, q, σ, s') implies the ℓ-structured LWE with parameters (n, m, q, σ, s, k).

Proof. We show how to transform the components in an ℓ'-succinct LWE instance into those of an ℓ-structured LWE instance. Specifically, consider the components $(\mathbf{A}, \mathbf{U}, \mathbf{T}')$ in an ℓ'-succinct LWE instance with parameters (n, m, q, σ, s'):

$$\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m} \quad \text{and} \quad \mathbf{U} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n\ell' \times m} \quad \text{and} \quad \mathbf{T}' \leftarrow [\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}]_{s'}^{-1}(\mathbf{G}_{n\ell'}), \tag{4.1}$$

We show how to use these components to construct a tuple $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$ distributed according to the specification of an ℓ-structured LWE instance with parameters (n, m, q, σ, s, k):

$$\mathbf{A} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times m} \quad \text{and} \quad \mathbf{Z} \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{n \times mk} \quad \text{and} \quad \mathbf{R} \leftarrow D_{\mathbb{Z},s}^{m \times \ell} \quad \text{and} \quad \mathbf{T} \leftarrow (\mathbf{V}_{\ell,k})_s^{-1}(\mathbf{G}_{n\ell}), \tag{4.2}$$

where $\mathbf{V}_{\ell,k}$ is the matrix from Eq. (3.1). We construct a reduction algorithm $\mathcal{R}$ as follows:

1. On input $(\mathbf{A}, \mathbf{U}, \mathbf{T}')$, parse

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{\ell'} \end{bmatrix} \in \mathbb{Z}_q^{n\ell \times m}, \tag{4.3}$$

   where $\mathbf{U}_i \in \mathbb{Z}_q^{n \times m}$. Next, write the gadget matrix $\mathbf{G}_{n\ell}$ as

$$\mathbf{G}_{n\ell} = \begin{bmatrix} \mathbf{v}_{1,1} & \cdots & \mathbf{v}_{1,\ell'} \\ \vdots & \ddots & \vdots \\ \mathbf{v}_{\ell,1} & \cdots & \mathbf{v}_{\ell,\ell'} \end{bmatrix} \in \mathbb{Z}_q^{n\ell \times \ell'}, \tag{4.4}$$

   where $\mathbf{v}_{i,j} \in \mathbb{Z}_q^n$ for all $i \in [\ell]$ and $j \in [\ell']$. For each $i \in [\ell]$, define

$$\hat{\mathbf{v}}_i \coloneqq \begin{bmatrix} \mathbf{v}_{i,1} \\ \vdots \\ \mathbf{v}_{i,\ell'} \end{bmatrix} \in \mathbb{Z}_q^{n\ell'}.$$

   Then, for each $i \in [\ell]$, sample

$$\begin{bmatrix} \mathbf{x}_{i,1} \\ \vdots \\ \mathbf{x}_{i,\ell'} \\ \mathbf{r}_i \end{bmatrix} \leftarrow \mathsf{SamplePre}\big([\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}], \mathbf{T}', \hat{\mathbf{v}}_i, s\big) \in \mathbb{Z}_q^{m\ell'+m}, \tag{4.5}$$

   where $\mathbf{x}_{i,j}, \mathbf{r}_i \in \mathbb{Z}_q^m$.

2. Next, sample $(\mathbf{Z}', \mathbf{T}_{\mathbf{Z}'}) \leftarrow \mathsf{TrapGen}(1^{nm}, q, k)$ and

$$\mathbf{d}_j \leftarrow \mathsf{SamplePre}(\mathbf{Z}', \mathbf{T}_{\mathbf{Z}'}, \mathsf{vec}(-\mathbf{U}_j), s) \in \mathbb{Z}_q^k \tag{4.6}$$

   for each $j \in [\ell']$. Here, $\mathsf{vec}(-\mathbf{U}_i) \in \mathbb{Z}_q^{nm}$ denotes the vectorization of $-\mathbf{U}_i$ (i.e., the vector obtained by vertically stacking the columns of $-\mathbf{U}_i$ from left to right).

3. For each $i \in [k]$, let $\mathbf{Z}_i \in \mathbb{Z}_q^{n \times m}$ be the matrix where $\mathsf{vec}(\mathbf{Z}_i) = \mathbf{z}'_i$ and $\mathbf{z}'_i \in \mathbb{Z}_q^{nm}$ is the i^{th} column of $\mathbf{Z}'$. Let $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \in \mathbb{Z}_q^{n \times mk}$ and $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \in \mathbb{Z}_q^{m \times \ell}$. Finally, let

$$\mathbf{V}_{\ell,k} = \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_\ell) \end{array}\right] \in \mathbb{Z}_q^{n\ell \times (m\ell + k)} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,\ell'} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{\ell,1} & \cdots & \mathbf{x}_{\ell,\ell'} \\ \mathbf{d}_1 & \cdots & \mathbf{d}_{\ell'} \end{bmatrix} \in \mathbb{Z}_q^{(m\ell+k) \times \ell'}. \tag{4.7}$$

   Output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.

We now show that if the inputs $(\mathbf{A}, \mathbf{U}, \mathbf{T}')$ to the above algorithm are distributed according to Eq. (4.1), then the outputs $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$ are statistically close to the distribution in Eq. (4.2). To do so, we define the following distributions:

- Hyb_0: In this experiment, the challenger first samples $(\mathbf{A}, \mathbf{U}, \mathbf{T}')$ according to Eq. (4.1) and then outputs $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T}) \leftarrow \mathcal{R}(\mathbf{A}, \mathbf{U}, \mathbf{T}')$. Specifically, the challenger does the following:
  - Sample $\mathbf{A} \xleftarrow{\text{R}} \mathbb{Z}_q^{n \times m}, \mathbf{U} \xleftarrow{\text{R}} \mathbb{Z}_q^{n\ell' \times m}$ and $\mathbf{T}' \leftarrow [\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}]_{s'}^{-1}(\mathbf{G}_{n\ell'})$. Parse $\mathbf{U}$ into $\mathbf{U}_1, \ldots, \mathbf{U}_{\ell'} \in \mathbb{Z}_q^{n \times m}$ according to Eq. (4.3).
  - For each $i \in [\ell]$, sample

$$\begin{bmatrix} \mathbf{x}_{i,1} \\ \vdots \\ \mathbf{x}_{i,\ell'} \\ \mathbf{r}_i \end{bmatrix} \leftarrow \mathsf{SamplePre}\big([\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}], \mathbf{T}', \hat{\mathbf{v}}_i, s\big) \in \mathbb{Z}_q^{m\ell'+m}.$$

  - Sample $(\mathbf{Z}', \mathbf{T}_{\mathbf{Z}'}) \leftarrow \mathsf{TrapGen}(1^{nm}, q, k)$ and $\mathbf{d}_j \leftarrow \mathsf{SamplePre}(\mathbf{Z}', \mathbf{T}_{\mathbf{Z}'}, \mathsf{vec}(-\mathbf{U}_j), s) \in \mathbb{Z}_q^k$ for each $j \in [\ell']$.
  - For each $i \in [k]$, let $\mathbf{Z}_i \in \mathbb{Z}_q^{n \times m}$ be the matrix where $\mathsf{vec}(\mathbf{Z}_i) = \mathbf{z}'_i$ and $\mathbf{z}'_i \in \mathbb{Z}_q^{nm}$ is the i^{th} column of $\mathbf{Z}'$. Let $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \in \mathbb{Z}_q^{n \times mk}$ and $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \in \mathbb{Z}_q^{m \times \ell}$.
  - Construct $\mathbf{V}_{\ell,k}$ and $\mathbf{T}$ according to Eq. (4.7) and output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.
- Hyb_1: Same as Hyb_0 except for each $i \in [\ell]$, the challenger changes how it samples $\mathbf{x}_{i,j}$ and $\mathbf{r}_i$:
  - For each $i \in [\ell]$, sample

$$\begin{bmatrix} \mathbf{x}_{i,1} \\ \vdots \\ \mathbf{x}_{i,\ell'} \\ \mathbf{r}_i \end{bmatrix} \leftarrow [\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}]_s^{-1}(\hat{\mathbf{v}}_i) \quad \text{where} \quad \hat{\mathbf{v}}_i = \begin{bmatrix} \mathbf{v}_{i,1} \\ \vdots \\ \mathbf{v}_{i,\ell'} \end{bmatrix} \in \mathbb{Z}_q^{n\ell'}. \tag{4.8}$$

- Hyb_2: Same as Hyb_1 except for each $i \in [\ell]$, the challenger again changes how it samples $\mathbf{x}_{i,j}$ and $\mathbf{r}_i$:
  - For each $i \in [\ell]$, sample $\mathbf{r}_i \leftarrow D_{\mathbb{Z},s}^m$ and for all $j \in [\ell']$, $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} - \mathbf{U}_j \mathbf{r}_i)$.
- Hyb_3: Same as Hyb_2 except the challenger changes how it samples $\mathbf{Z}'$ and $\mathbf{d}_j$.

- Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{n\ell'\times m}$ and $\mathbf{T}' \leftarrow [\mathbf{I}_{\ell'} \otimes \mathbf{A} \mid \mathbf{U}]_{s'}^{-1}(\mathbf{G}_{n\ell'})$. Parse $\mathbf{U}$ into $\mathbf{U}_1, \ldots, \mathbf{U}_{\ell'} \in \mathbb{Z}_q^{n\times m}$ according to Eq. (4.3).
- For each $i \in [\ell]$, sample $\mathbf{r}_i \leftarrow D_{\mathbb{Z},s}^m$ and for all $j \in [\ell']$, $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i)$.
- Sample $\mathbf{Z}' \xleftarrow{R} \mathbb{Z}_q^{nm\times k}$ and $\mathbf{d}_j \leftarrow (\mathbf{Z}')_s^{-1}(\mathrm{vec}(-\mathbf{U}_j))$ for each $j \in [\ell']$.
- For each $i \in [k]$, let $\mathbf{Z}_i \in \mathbb{Z}_q^{n\times m}$ be the matrix where $\mathrm{vec}(\mathbf{Z}_i) = \mathbf{z}'_i$ and $\mathbf{z}'_i \in \mathbb{Z}_q^{nm}$ is the i^{th} column of $\mathbf{Z}'$. Let $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \in \mathbb{Z}_q^{n\times mk}$ and $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \in \mathbb{Z}_q^{m\times \ell}$.
- Construct $\mathbf{V}_{\ell,k}$ and $\mathbf{T}$ according to Eq. (4.7) and output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.

- Hyb_4: Same as Hyb_3, except the challenger reorders some of the sampling steps:
  - Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \xleftarrow{R} \mathbb{Z}_q^{n\times mk}$ and set $\mathbf{Z}' = [\mathrm{vec}(\mathbf{Z}_1) \mid \cdots \mid \mathrm{vec}(\mathbf{Z}_k)]$. Sample $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \leftarrow D_{\mathbb{Z},s}^{m\times\ell}$.
  - Sample $\mathbf{U} \xleftarrow{R} \mathbb{Z}_q^{n\ell'\times m}$. Parse $\mathbf{U}$ into $\mathbf{U}_1, \ldots, \mathbf{U}_{\ell'} \in \mathbb{Z}_q^{n\times m}$ according to Eq. (4.3). Sample $\mathbf{d}_j \leftarrow (\mathbf{Z}')_s^{-1}(\mathrm{vec}(-\mathbf{U}_j))$ for each $j \in [\ell']$.
  - For each $i \in [\ell]$ and $j \in [\ell']$, sample $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i)$.
  - Construct $\mathbf{V}_{\ell,k}$ and $\mathbf{T}$ according to Eq. (4.7) and output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.
- Hyb_5: Same as Hyb_4, except the challenger changes how it samples $\mathbf{d}_j$ and $\mathbf{U}_j$:
  - Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \xleftarrow{R} \mathbb{Z}_q^{n\times mk}$ and set $\mathbf{Z}' = [\mathrm{vec}(\mathbf{Z}_1) \mid \cdots \mid \mathrm{vec}(\mathbf{Z}_k)]$. Sample $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \leftarrow D_{\mathbb{Z},s}^{m\times\ell}$.
  - For each $j \in [\ell']$, sample $\mathbf{d}_j \leftarrow D_{\mathbb{Z},s}^k$, and set $\mathbf{U}_j \in \mathbb{Z}_q^{n\times m}$ to be the matrix where $\mathrm{vec}(-\mathbf{U}_j) = \mathbf{Z}'\mathbf{d}_j$.
  - For each $i \in [\ell]$ and $j \in [\ell']$, sample $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i)$.
  - Construct $\mathbf{V}_{\ell,k}$ and $\mathbf{T}$ according to Eq. (4.7) and output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.
- Hyb_6: In this experiment, the challenger changes how it samples $\mathbf{T}$.
  - Sample $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}$, $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k] \xleftarrow{R} \mathbb{Z}_q^{n\times mk}$ and $\mathbf{R} = [\mathbf{r}_1 \mid \cdots \mid \mathbf{r}_\ell] \leftarrow D_{\mathbb{Z},s}^{m\times\ell}$.
  - Construct $\mathbf{V}_{\ell,k}$ according to Eq. (4.7) and sample $\mathbf{T} \leftarrow (\mathbf{V}_{\ell,k})_s^{-1}(\mathbf{G}_{n\ell})$.
  - Output $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$.

  In this experiment, the challenger samples $(\mathbf{A}, \mathbf{Z}, \mathbf{R}, \mathbf{T})$ according to Eq. (4.2).

We show that the outputs of each pair of adjacent hybrids are statistically indistinguishable.

Lemma 4.2. *Suppose $m \geq 2n\log q$, $s' \geq \log m$, $s \geq m^{3/2}(\ell'+1)s'\log(n\ell')$, and $\ell' = \mathrm{poly}(n)$. Then, $\mathsf{Hyb}_0 \overset{s}{\approx} \mathsf{Hyb}_1$.*

Proof. Since $m \geq 2n\log q$, $s' \geq \log m$, and $\ell' = O(n\log q) = \mathrm{poly}(\lambda)$, we can appeal to Lemmas 2.3 and 2.5 together with a union bound to conclude that with probability $1 - \mathrm{negl}(\lambda)$, $\|\mathbf{T}'\| \leq \sqrt{m}s'$. Next, if $s \geq m^{3/2}(\ell'+1)s'\log(n\ell') > m(\ell'+1)\|\mathbf{T}'\|\log(n\ell')$, then the claim follows by Lemma 2.6. □

Lemma 4.3. *If $m \geq 2n\log q$, q is prime, and $s \geq \log(m\ell')$, then $\mathsf{Hyb}_1 \overset{s}{\approx} \mathsf{Hyb}_2$.*

Proof. Follows immediately by Lemma 2.5. □

Lemma 4.4. *If $k \geq 3nm \log q$ and $s \geq k \log(nm)$, then* $\mathsf{Hyb}_2 \stackrel{s}{\approx} \mathsf{Hyb}_3$.

Proof. Since $k \geq 3nm \log q$, the distribution of $\mathbf{Z}'$ in Hyb_2 is $\mathsf{negl}(\lambda)$-close to uniform (over $\mathbb{Z}_q^{nm \times k}$) by Lemma 2.6, which is the distribution of $\mathbf{Z}'$ in Hyb_3. Moreover, since $s \geq k \log(nm)$, by Lemma 2.6, the distribution of $\mathbf{d}_j$ in Eq. (4.6) is statistically close to sampling $\mathbf{d}_j \leftarrow (\mathbf{Z}')_s^{-1}(\mathsf{vec}(-\mathbf{U}_j))$. □

Lemma 4.5. Hyb_3 *and* Hyb_4 *are identically distributed.*

Proof. In both experiments, the distribution of $\mathbf{Z}$ is uniform over $\mathbb{Z}_q^{n \times mk}$ and $\mathbf{Z}' = [\mathsf{vec}(\mathbf{Z}_1) \mid \cdots \mid \mathsf{vec}(\mathbf{Z}_k)]$. Similarly, in both experiments, the distribution of $\mathbf{R}$ is $D_{\mathbb{Z},s}^{m \times \ell}$. All other quantities are sampled identically in the two experiments, just in a different order. □

Lemma 4.6. *If $k \geq 2nm \log q$, q is prime, and $s \geq \log k$, then* $\mathsf{Hyb}_4 \stackrel{s}{\approx} \mathsf{Hyb}_5$.

Proof. The only difference between Hyb_4 and Hyb_5 is the distribution of $\mathbf{U}_j$ and $\mathbf{d}_j$ for each $j \in [\ell']$. This follows by Lemma 2.4. Namely, for every $j \in [\ell']$, the following distributions are statistically indistinguishable:

- Sample $\mathbf{Z}' \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{nm \times k}$, $\mathbf{w}_j \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{nm}$, and $\mathbf{d}_j \leftarrow (\mathbf{Z}')_s^{-1}(\mathbf{w}_j)$, and output $(\mathbf{Z}', \mathbf{d}_j, \mathbf{w}_j)$.
- Sample $\mathbf{Z}' \xleftarrow{\mathrm{R}} \mathbb{Z}_q^{nm \times k}$, $\mathbf{d}_j \leftarrow D_{\mathbb{Z},s}^k$, and $\mathbf{w}_j = \mathbf{Z}'\mathbf{d}_j$. Output $(\mathbf{Z}', \mathbf{d}_j, \mathbf{w}_j)$.

Setting $\mathbf{U}_j \in \mathbb{Z}_q^{n \times m}$ to be the matrix where $\mathsf{vec}(-\mathbf{U}_j) = \mathbf{w}_j$, the first distribution corresponds to Hyb_4 while the second corresponds to Hyb_5. Since $\ell' = \mathsf{poly}(n)$, the claim follows by a hybrid argument. □

Lemma 4.7. *If $m \geq 2n \log q$, q is prime, and $s \geq \log(m\ell)$, then* $\mathsf{Hyb}_5 \stackrel{s}{\approx} \mathsf{Hyb}_6$.

Proof. In Hyb_5, the challenger sets $\mathbf{U}_j$ such that $\mathsf{vec}(-\mathbf{U}_j) = \mathbf{Z}'\mathbf{d}_j$ and then samples $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i)$ for each $i \in [\ell]$ and $j \in [\ell']$. Since $m \geq 2n \log q$, and $\mathbf{A}$ is sampled uniformly from $\mathbb{Z}_q^{n \times m}$, the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$ by Corollary 2.2 with overwhelming probability. Thus, $\mathbf{A}\mathbf{x}_{i,j} = \mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i$. Next, $\mathbf{Z}' = [\mathsf{vec}(\mathbf{Z}_1) \mid \cdots \mid \mathsf{vec}(\mathbf{Z}_k)]$, so

$$\mathbf{Z}(\mathbf{d}_j \otimes \mathbf{I}_m) = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_k](\mathbf{d}_j \otimes \mathbf{I}_m) = -\mathbf{U}_j. \tag{4.9}$$

By Eq. (2.1), for all $i \in [\ell]$ and $j \in [\ell']$, we can write

$$\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)\mathbf{d}_j = \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)(\mathbf{d}_j \otimes 1) = \mathbf{Z}(\mathbf{d}_j \otimes \mathbf{I}_m)\mathbf{r}_i = -\mathbf{U}_j\mathbf{r}_i.$$

In particular, this means that for all $i \in [\ell]$ and $j \in [\ell']$,

$$\mathbf{A}\mathbf{x}_{i,j} - \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)\mathbf{d}_j = \mathbf{v}_{i,j} - \mathbf{U}_j\mathbf{r}_i + \mathbf{U}_j\mathbf{r}_i = \mathbf{v}_{i,j}. \tag{4.10}$$

We now claim that $\mathbf{T}$ is a gadget trapdoor for $\mathbf{V}_{\ell,k}$. From Eqs. (4.4), (4.7) and (4.10),

$$\mathbf{V}_{\ell,k}\mathbf{T} = \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_1) \\ & \ddots & & \vdots \\ & & \mathbf{A} & -\mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_\ell) \end{array}\right] \cdot \begin{bmatrix} \mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,\ell'} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{\ell,1} & \cdots & \mathbf{x}_{\ell,\ell'} \\ \mathbf{d}_1 & \cdots & \mathbf{d}_{\ell'} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{1,1} & \cdots & \mathbf{v}_{1,\ell'} \\ \vdots & \ddots & \vdots \\ \mathbf{v}_{\ell,1} & \cdots & \mathbf{v}_{\ell,\ell'} \end{bmatrix} = \mathbf{G}_{n\ell}.$$

Since the challenger in Hyb_5 samples $\mathbf{d}_j \leftarrow D_{\mathbb{Z},s}^k$ and $\mathbf{x}_{i,j} \leftarrow \mathbf{A}_s^{-1}(\mathbf{v}_{i,j} + \mathbf{Z}(\mathbf{I}_k \otimes \mathbf{r}_i)\mathbf{d}_j)$ for all $i \in [\ell]$ and $j \in [\ell']$, by Lemma 2.5, the distribution of $\mathbf{T}$ is statistically close to $\mathbf{T} \leftarrow (\mathbf{V}_{\ell,k})_s^{-1}(\mathbf{G}_{n\ell})$. This is the distribution in Hyb_6. □

Theorem 4.1 now follows by combining Lemmas 4.3 to 4.7. Specifically, the reduction algorithm $\mathcal{R}$ shows how to efficiently simulate the components of the ℓ-structured LWE assumption using the components from the ℓ'-succinct LWE assumption. □

SIS variants. Our proof for Theorem 4.1 shows how to use a trapdoor for the ℓ'-succinct LWE assumption to construct a trapdoor for the ℓ-structured LWE assumption. The same transformation would apply to show a similar reduction between the ℓ'-succinct SIS assumption and the ℓ-structured SIS assumption. We can also construct an analogous reduction algorithm that transforms an instance of the ℓ'-structured LWE assumption into an instance of the ℓ-succinct LWE assumption. Thus, up to a $O(n \log q)$ increase in the dimension (and polynomial blow-up in the noise parameters), these assumptions are essentially equivalent.

Acknowledgments

We thank Hoeteck Wee for many insightful discussions on distributed broadcast encryption and the ℓ-succinct LWE assumption. We thank Brent Waters and the anonymous TCC reviewers for providing many helpful comments on the presentation. David J. Wu is supported by NSF CNS-2140975, CNS-2318701, a Microsoft Research Faculty Fellowship, and a Google Research Scholar award.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, 2010.

[ACL+22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In *CRYPTO*, 2022.

[AFLN24] Martin R. Albrecht, Giacomo Fenzi, Oleksandra Lapiha, and Ngoc Khanh Nguyen. SLAP: succinct lattice-based polynomial commitments from standard assumptions. In *EUROCRYPT*, 2024.

[AGHS13] Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete gaussian leftover hash lemma over infinite domains. In *ASIACRYPT*, 2013.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, 1996.

[Alb24] Martin R. Albrecht. SIS with hints zoo, 2024.

[AR16] Divesh Aggarwal and Oded Regev. A note on discrete gaussian combinations of lattice vectors. *Chic. J. Theor. Comput. Sci.*, 2016, 2016.

[AT24] Nuttapong Attrapadung and Junichi Tomida. A modular approach to registered abe for unbounded predicates. In *CRYPTO*, 2024.

[BCFL23] David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Chainable functional commitments for unbounded-depth circuits. In *TCC*, 2023.

[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 *EUROCRYPT*, 2014.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, 2001.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, 2005.

[BLM+24] Pedro Branco, Russell W. F. Lai, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Ivy K. Y. Woo. Traitor tracing without trusted authority from registered functional encryption. *IACR Cryptol. ePrint Arch.*, 2024.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In *CRYPTO*, 2013.

[BV22] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. In *ITCS*, 2022.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *CRYPTO*, 2014.

[CES21] Kelong Cong, Karim Eldefrawy, and Nigel P. Smart. Optimizing registration based encryption. In *Cryptography and Coding*, 2021.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, 2010.

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In *CRYPTO*, 2023.

[DKL+23] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In *EUROCRYPT*, 2023.

[DKW21] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority ABE for dnfs from LWE. In *EUROCRYPT*, 2021.

[DPY23] Pratish Datta, Tapas Pal, and Shota Yamada. Registered FE beyond predicates:(attribute-based) linear functions and more. *Cryptology ePrint Archive*, 2023.

[FFM+23] Danilo Francati, Daniele Friolo, Monosij Maitra, Giulio Malavolta, Ahmadreza Rahimi, and Daniele Venturi. Registered (inner-product) functional encryption. In *ASIACRYPT*, 2023.

[FKdP23] Dario Fiore, Dimitris Kolonelos, and Paola de Perthuis. Cuckoo commitments: Registration-based encryption and key-value map commitments for large spaces. In *ASIACRYPT*, 2023.

[FMN23] Giacomo Fenzi, Hossein Moghaddas, and Ngoc Khanh Nguyen. Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency. *IACR Cryptol. ePrint Arch.*, 2023.

[FN93] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, 1993.

[FWW23] Cody Freitag, Brent Waters, and David J. Wu. How to use (plain) witness encryption: Registered ABE, flexible broadcast, and more. In *CRYPTO*, 2023.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.

[GHM+19] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. Registration-based encryption from standard assumptions. In *PKC*, 2019.

[GHMR18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: Removing private-key generator from IBE. In *TCC*, 2018.

[GKMR23] Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. Efficient registration-based encryption. In *ACM CCS*, 2023.

[GKPW24] Sanjam Garg, Dimitris Kolonelos, Guru-Vamsi Policharla, and Mingyuan Wang. Threshold encryption with silent setup. In *CRYPTO*, 2024.

[GKW18] Romain Gay, Lucas Kowalczyk, and Hoeteck Wee. Tight adaptively secure broadcast encryption with short ciphertexts and keys. In *SCN*, 2018.

[GLWW23] Rachit Garg, George Lu, Brent Waters, and David J. Wu. Realizing flexible broadcast encryption: How to broadcast to a public-key directory. In *ACM CCS*, 2023.

[GLWW24] Rachit Garg, George Lu, Brent Waters, and David J. Wu. Reducing the CRS size in registered ABE systems. In *CRYPTO*, 2024.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[GV20] Rishab Goyal and Satyanarayana Vusirikala. Verifiable registration-based encryption. In *CRYPTO*, 2020.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC*, 2013.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *EUROCRYPT*, 2009.

[HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *FOCS*, 2023.

[HLWW23] Susan Hohenberger, George Lu, Brent Waters, and David J. Wu. Registered attribute-based encryption. In *EUROCRYPT*, 2023.

[KMW23] Dimitris Kolonelos, Giulio Malavolta, and Hoeteck Wee. Distributed broadcast encryption from bilinear groups. In *ASIACRYPT*, 2023.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, 2012.

[PPS12] Duong Hieu Phan, David Pointcheval, and Mario Strefler. Decentralized dynamic broadcast encryption. In *SCN*, 2012.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *CRYPTO*, 2022.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-io from evasive LWE. In *ASIACRYPT*, 2022.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In *EUROCRYPT*, 2022.

[Wee24] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In *CRYPTO*, 2024.

[WQZDF10] Qianhong Wu, Bo Qin, Lei Zhang, and Josep Domingo-Ferrer. Ad hoc broadcast encryption. In *ACM CCS*, 2010.

[WW23a] Hoeteck Wee and David J. Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. In *ASIACRYPT*, 2023.

[WW23b] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In *EUROCRYPT*, 2023.

[WWW22] Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In *TCC*, 2022.

[ZZGQ23] Ziqi Zhu, Kai Zhang, Junqing Gong, and Haifeng Qian. Registered ABE via predicate encodings. In *ASIACRYPT*, 2023.