

Scalable Registration-Based Encryption from Lattices

Michael Klooß^{*}, Russell W. F. Lai[†], Jan Niklas Siemer[‡], and Monisha Swarnakar[†]

^{*}Karlsruhe Institute of Technology & KASTEL Security Research Labs, Germany

[†]Aalto University, Finland

[‡]King's College London, United Kingdom

Abstract

Registration-Based Encryption (RBE) is a public-key encryption mechanism which allows a user to register their identity (e.g. email address) and self-generated public key with a key curator (e.g. an organisation). The key curator aggregates these keys into a compact digest. Using only this digest and the recipient's identity, anyone can encrypt messages to any registered user. As the key curator is not entrusted with any secrets, RBE presents a solution to the key escrow problem, which impedes the adoption of Identity-Based Encryption. This makes RBE an attractive solution for secure communication with and among members of an organisation while preserving user privacy. Despite recent advances [Döttling-Kolonelos-Lai-Lin-Malavolta-Rahimi, EUROCRYPT'23; Fiore-Kolonelos-de-Perthuis, ASIACRYPT'23], practical constructions of RBE are still limited to a small number of registered users (e.g. 1024), lack post-quantum security, or have ciphertext sizes scaling in the order of GB.

The predominant way towards constructing practical RBE is a generic transformation from Laconic Encryption (LE). In this work, we identify an efficiency bottleneck in this transformation and present a new primitive called Batched Laconic Encryption (BLE) which admits a more succinct transformation to RBE. Our resulting RBE scheme is the first post-quantum construction that simultaneously supports a large number of registered users and asymptotically outperforms all comparable RBE schemes. Concretely, for at most 2^{30} registered users at 128-bit security, our scheme achieves a ciphertext size of 7 MB, improving on previously reported results by three orders of magnitude. We confirm our results through an open-source prototype implementation demonstrating that all algorithms execute within a few milliseconds. The post-quantum security of our construction is based on the standard Learning with Errors assumption, and our analysis enables several tweaks to significantly reduce ciphertext sizes in practical deployments.

1. Introduction

In today's highly digitalised society, it is common to exchange emails with dozens of other members belonging to the same or different organisations on a daily basis. Encrypting all these emails with an ordinary public-key encryption (PKE) would introduce a key-management problem – the sender needs to keep track of the most recent public key of all potential recipients.

Identity-based encryption (IBE) [1] was proposed as a solution to the key-management problem. Using an IBE, each organisation issues identity-based secret keys to its members. The sender can encrypt a message to a member of an organisation by knowing the public key of the organisation and the identity (e.g. name or email address) of the recipient. This solution requires management of one public key per organisation, which is easier than handling hundreds or thousands of individual public keys. Clearly, the applications of IBE go beyond only secure email communication. However, despite the promises and abundance of practical IBE schemes (e.g. [2], [3]), the real-world adoption of IBE remains limited, potentially due to the key escrow problem – the organisation can decrypt all emails sent to its members.

Registration-based Encryption (RBE). Registration-based encryption (RBE) was introduced by Garg et al. [4] as a decentralised alternative to identity-based encryption (IBE) which removes the trusted key authority and hence the notorious key-escrow problem. With an RBE, any user can generate their own public and secret keys, as in the PKE-based solution, and register their public key associated to their identity with a key curator (e.g. an organisation). The task of the key curator is only to maintain a registry of public keys, which can be publicly audited. The key curator also publishes and maintains a small state st , which contains e.g. a short digest of the registry, against which anyone can encrypt a message to any registered identity without knowing the corresponding public key. The recipient can then decrypt the ciphertext using their own secret key and a (publicly computable) membership witness obtained from the key curator. In the context of encrypted emails, the sender only needs to keep track of the state published by each organisation.

A key feature of an RBE is that the public parameters pp and state st are compact, i.e. they are of size sublinear in the maximum number of registered users N , which could range from e.g. 2^{10} for a medium-sized organisation to 2^{30} for popular email service providers. Furthermore, RBE enforces the efficiency requirement that any single registered user only needs to update their membership witness at most $\tilde{O}(\log N)$ times throughout the lifetime of the system.

	Setting	$\mathcal{ID}$	Compactness	$ \text{ctxt} $	$ \text{pp} + \text{st} $
[10]	Pairing	$\{0, 1\}^*$	Adaptive	$\mathcal{O}(\lambda^2 \cdot \log N)$	$\mathcal{O}\left(\lambda^2 \cdot N^{\frac{2}{3}} \cdot \log N\right)$
[7]	Pairing	$[N]$	Adaptive	$\mathcal{O}(\lambda \cdot \log N)$	$\mathcal{O}\left(\lambda \cdot N^{\frac{1}{2}} \cdot \log N\right)$
[9, P1]	Pairing	$\{0, 1\}^*$	Adaptive	$\mathcal{O}(\lambda^2 \cdot \log N)$	$\mathcal{O}\left(\lambda^{\frac{3}{2}} \cdot N^{\frac{1}{2}} \log N\right)$
[9, P2]	Pairing	$\{0, 1\}^*$	Selective	$\mathcal{O}(\lambda \cdot \log N)$	$\mathcal{O}\left(\lambda \cdot N^{\frac{1}{2}} \cdot \log N\right)$
[8]	Lattice	$\{0, 1\}^*$	Adaptive	$\tilde{\mathcal{O}}(\lambda^2) \cdot \log N$	$\tilde{\mathcal{O}}(\lambda) \cdot \log N$
[9]	Lattice	$\{0, 1\}^*$	Selective	$\tilde{\mathcal{O}}(\lambda) \cdot \log^2 N$	$\tilde{\mathcal{O}}(\lambda) \cdot \log N$
This work	Lattice	$\{0, 1\}^*$	Adaptive	$\tilde{\mathcal{O}}(\lambda^2) + \tilde{\mathcal{O}}(\lambda) \cdot \log N$	$\tilde{\mathcal{O}}(\lambda) \cdot \log N$

Table 1: Comparison of algebraic constructions of RBE with sizes in bits, extending [9, Table 1].

Existing RBE Constructions. Early constructions of RBE relied on obfuscation [4] or recursive garbled circuits [5], which were mostly theoretical even with optimisations [6]. Recent works [7], [8], [9] proposed fully algebraic constructions¹ with near-practical efficiency albeit with various limitations, as summarised in Table 1.

The pairing-based scheme of [7] achieves a ciphertext size of $\mathcal{O}(\lambda \log N)$ but restricts the identity space $\mathcal{ID}$ to $[N]$ with an a priori bounded N . This forbids using 128-bit or 256-bit hash digests of ‘real’, arbitrarily long identity strings as ‘logical’ identities.

The first pairing-based scheme in [9] lifts this restriction, i.e. it allows identities to be arbitrary finite strings, at the cost of having $\mathcal{O}(\lambda^2 \log N)$ size ciphertexts. Their second pairing-based scheme brings the ciphertext size back down to $\mathcal{O}(\lambda \log N)$, or concretely 1.67 MB for $N = 1000$. However, this comes at the cost of achieving only a selective notion of compactness, meaning that an adversary could adaptively register identities to make the state size grow with $\Omega(N) \cdot \text{poly}(\lambda)$, in a sense mounting a denial-of-service attack. Finally, all pairing-based constructions are not post-quantum secure, require trusted setups, and have public parameters and state sizes $\sqrt{N} \cdot \text{poly}(\lambda)$. Requiring a trusted setup, while better than key escrow as in IBEs, is arguably against the core philosophy of the registration-based setting of eliminating trust from any authority.

For plausibly post-quantum secure schemes, the lattice-based scheme of Döttling, Kolonelos, Lai, Lin, Malavolta and Rahimi (DKLLMR) [8] achieves a ciphertext size of $\tilde{\mathcal{O}}(\lambda^2) \cdot \log N$, which is concretely estimated in [9] to be 2.4 GB for $N = 1024$ users. For $N = 2^{30}$, the ciphertext size grows to a staggering 7.2 GB.² The ciphertext size is improved in the lattice-based scheme of [9] to $\tilde{\mathcal{O}}(\lambda) \cdot \log^2 N$ but at the cost of selective compactness. Both lattice-based constructions have transparent setups, meaning that the public parameters consist of uniformly random values that can be simulated with a hash function (modelled as a random oracle), and have public parameters and digests sizes $\tilde{\mathcal{O}}(\lambda) \cdot \log N$.

To summarise, if we insist on logarithmic-size public parameters and digests and adaptive compactness, the best known scheme remains DKLLMR, in which ciphertext sizes are in the order of GB.

1.1. Our Contributions

Scalable RBE. In this work, we propose a new lattice-based construction of RBE based on the standard Learning with Errors (LWE) assumption with practical efficiency even when scaling to billions of users. Asymptotically, our scheme strictly improves DKLLMR [8], reducing the ciphertext size from $\tilde{\mathcal{O}}(\lambda^2 \log N)$ to

$$\tilde{\mathcal{O}}(\lambda^2) + \tilde{\mathcal{O}}(\lambda) \cdot \log N$$

which is significantly less sensitive in N . Concretely, our ciphertext size is 7.0 MB for $N = 2^{30}$ compared to 7.2 GB of DKLLMR. Due to the insensitivity in N , the ciphertext size does not vary greatly even if N were to grow to 2^{128} , indicating a ciphertext size of 7.1 MB. For a more detailed comparison with existing schemes, we refer to Table 1.

Batched Laconic Encryption. Our RBE construction follows an established framework of generically transforming a laconic encryption (LE) scheme to an RBE (see e.g. [8]). All existing near-practical RBEs are built via such a transformation. To recall, an LE is essentially an RBE without the efficiency requirement on membership witness updates.³ In a nutshell, the transformation involves arranging a maximum number of $N = 2^B$ registered RBE users into B buckets. Each bucket is

¹We omit constructions of more advanced primitives such as registered attribute-based encryption, which are even more computationally expensive.

²Note that $1024 = 2^{10}$, meaning that a DKLLMR RBE ciphertext consists of 10 laconic encryption (LE) ciphertexts. For $N = 2^{30}$, the number of LE ciphertexts becomes 30, and $7.2 = 2.4 \cdot \frac{30}{10}$.

³As with RBE, for LE, a key curator computes a short digest of all public keys of registered identities. Then, anyone can encrypt to (registered) identities given the digest. However, the membership witness for an identity may (and typically does) change on every registration.

assigned to one LE instance so that an RBE ciphertext consists of B LE ciphertexts. If each LE ciphertext is of size S , then the total ciphertext size would be $B \cdot S$.

The technical ingredient enabling our efficiency improvement is a new notion which we call batched laconic encryption (BLE). A BLE allows an encryptor to encrypt to an identity registered in a subset of B LE instances with an additive instead of multiplicative overhead in B . More precisely, our BLE ciphertext size will be $S + B \cdot S'$ for some $S' \ll S$.

Our BLE construction is an extension and optimisation of the LE construction of DKLLMR. To turn the multiplicative overhead into an additive one, we observe that a large portion of a ciphertext can be reused across all B instances. While this observation alone suffices to achieve the asymptotic improvement, we also employ a number of optimisation techniques to further improve concrete efficiency. In general, we replace *statistical/exact* components with *computational/approximate* alternatives in several contexts, which result in significantly smaller dimensions and modulus size, which in turn leads to smaller key and ciphertext sizes. In Section 5, we discuss our optimisations in more detail.

Flexibility of BLE. RBE is optimised for reducing the number of times that a registered user needs to update their membership witness as more and more users join the system. All users are treated equally in that, typically, identities are strings of the same length and without any semantic meaning. However, in some applications it may be natural to support updating and removing the keys of registered users. Identity strings may be of varying lengths, have semantic meaning and exhibit hierarchies or groupings. Our notion of BLE is designed to be formally flexible enough for handling these situations. In particular, it does not require identities to be of the same length, and exposes the data structure of the registry: For any choice of the B sub-registries, a higher level application can instruct the BLE to insert or remove an identity-key tuple or update an existing key. In our BLE construction, the ciphertext is proportional to the identity length of the recipient, i.e. shorter identities have shorter ciphertexts. Thus, apart from being a better starting point for obtaining efficient RBEs, we believe that BLE is interesting in its own right as a primitive that offers greater flexibility to applications compared to RBE.

Implementation and Benchmarks. We implement our RBE construction in Rust with support for hardware-accelerated matrix multiplication using the number theoretic transform (NTT). As our encryption algorithm requires more than $1.7 \cdot 10^6$ samples from discrete Gaussian distributions, we implement discrete Gaussian samplers with high throughput rates. Additionally, we apply best-practices from Kyber [11] to our construction including lossy compression via bit-dropping for ciphertexts. With our chosen parameters, our scheme achieves 128-bit security, yields ciphertext sizes of 7.0 MB, and executes encryption and decryption in 15.56 ms and 10.43 ms, respectively, on a laptop.

1.2. Open Problem

Even though the RBE construction presented in this work employs all known theoretical improvements and practical optimisations that we are aware of, its ciphertext size is still in single-digit MB, i.e. 4 orders of magnitude larger than the size of a Kyber-512 ciphertext [11]. The reason is that our RBE ciphertext consists of $\mathcal{O}(\ell \log k)$ components if $[k]^\ell$ is the identity space. To achieve adaptive compactness [9], $\ell \log k$ must be large enough (i.e. $\Omega(\lambda)$) to be the output length of a collision resistant hash function. Removing this multiplicative dependency on λ could potentially bring the ciphertext size to double-digit KB territory. We believe that such an improvement requires fundamental changes to the construction, by replacing the vector commitment utilised in our construction, namely a Merkle tree [12], with a more succinct vector commitment [13]. Doing so while retaining the transparent setup and short public parameters appears to be a challenging theoretical problem, which we leave for future work.

1.3. Related Work

All RBE constructions achieving the standard security notion, including ours, require a setup. Indeed, Dujmović, Malavolta and Qian [14] showed that achieving standard RBE security in the plain model (i.e. without setup) is impossible. While they also showed how to circumvent this impossibility by assuming the existence of one honest registered user, the latter is essentially also a trusted setup. Fortunately, the setups in our RBE and the lattice-based schemes of [8], [9] are transparent, meaning that the public parameters consist of only uniformly random values. Since the latter can be generated by a hash function (formally modelled as a random oracle), a transparent setup is as good as no setup in practice.

Beyond RBE, a related line of work (e.g. [10], [15], [16], [17], [18], [19], [20], [21], [22], [23]) explores more expressive registered primitives, such as registered attribute-based encryption (RABE) and registered functional encryption (RFE), which extend RBE by enabling fine-grained access control over ciphertexts. However, these results so far have only demonstrated theoretical feasibility and are still far from practical.

2. Technical Overview

Our BLE and RBE constructions build on the LE scheme of Döttling, Kolonelos, Lai, Lin, Malavolta and Rahimi (DKLLMR) [8]. Therefore, we begin by providing a high-level overview of their LE construction, their generic LE-to-RBE transformation, and discuss the practical implications of their approach. Then, we introduce our core technical contribution, *Batched Laconic Encryption*, which allows us to achieve an improved BLE-to-RBE transformation with significantly less ciphertext size overhead compared to the LE-to-RBE transformation. We proceed to define our BLE construction implicitly by highlighting the differences from the LE construction of DKLLMR. Finally, we highlight further optimisations employed to enable more practical ciphertext sizes.

2.1. Laconic Encryption of DKLLMR

The DKLLMR LE scheme is based on a SIS-based Merkle tree [12], [24] and dual-Regev encryption [3]. Its public parameters consist of random matrices $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B} \leftarrow_{\$} \mathcal{R}_q^{n \times m}$ and a random vector $\mathbf{y}_{\dagger} \leftarrow_{\$} \mathcal{R}_q^n$ called the “terminator”. To generate a key pair, a user samples a short vector $\mathbf{x}_{\text{id}} \leftarrow_{\$} \mathcal{R}^m$ as the secret key and computes the public-key $\mathbf{y}_{\text{id}} = \mathbf{B}\mathbf{x}_{\text{id}} \bmod q$. The public-key is registered in the registry reg , which is structured as a depth ℓ binary Merkle tree defined with respect to hash $H(\mathbf{y}_0, \mathbf{y}_1) := -(\mathbf{A}_0\mathbf{G}^{-1}(\mathbf{y}_0) + \mathbf{A}_1\mathbf{G}^{-1}(\mathbf{y}_1)) \bmod q$ where $\mathbf{G}^{-1}$ denotes binary decomposition. Each public key $\mathbf{y}_{\text{id}}$ is registered in the leaf node indexed by the identity string id and the root label is computed by recursively labelling the next parent node with the hash value of the labels of its children. Any unlabelled child nodes on the Merkle-path to id are set to the terminator $\mathbf{y}_{\dagger}$. The root of the Merkle tree $\mathbf{y}_{\epsilon}$ is a short digest of all registered public keys and is stored in the state st . See Figure 1 for an example configuration of reg .

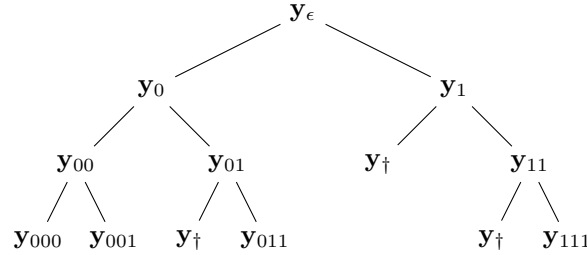


Figure 1: An example layout of the registry reg of the LE of DKLLMR [8] for registered identities $\{000, 001, 011, 111\}$.

To encrypt, the dual-Regev [3] encryption algorithm is employed w.r.t. the tuple $(\mathbf{A}_{\text{id}}, \mathbf{v})$ where

$$\mathbf{A}_{\text{id}} := \begin{pmatrix} \mathbf{A}_0 \parallel \mathbf{A}_1 & & & \\ \mathbf{i}_{\text{id}_0}^T \otimes \mathbf{G} & \ddots & & \\ & \ddots & \mathbf{A}_0 \parallel \mathbf{A}_1 & \\ & & \mathbf{i}_{\text{id}_{\ell-1}}^T \otimes \mathbf{G} & \mathbf{B} \end{pmatrix}, \mathbf{v} := \begin{pmatrix} \mathbf{y}_{\epsilon} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}.$$

More explicitly, a message $\mu \in \mathcal{R}_p$ is encrypted by sampling a uniformly random vector $\mathbf{r} \leftarrow_{\$} \mathcal{R}_q^{(\ell+1)n}$, short errors $\mathbf{e} \leftarrow_{\$} \mathcal{R}^{(2\ell+1)m}$, $\tilde{e} \leftarrow_{\$} \mathcal{R}$ and computing

$$\mathbf{c}^T := \mathbf{r}^T \mathbf{A}_{\text{id}} + \mathbf{e}^T \bmod q, \quad d := \mathbf{r}^T \mathbf{v} + \tilde{e} + \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu \bmod q.$$

Note that $\mathbf{A}_{\text{id}}$ is a block bi-diagonal matrix consisting of the “hash-blocks” $\mathbf{H}_{\text{id}_i} = (\mathbf{A}_0 \parallel \mathbf{A}_1; \mathbf{i}_{\text{id}_i} \otimes \mathbf{G})$, where $\mathbf{i}_{\text{id}_i} \otimes \mathbf{G}$ is a shorthand for $(1 - \text{id}_i) \cdot \mathbf{G} \parallel \text{id}_i \cdot \mathbf{G}$ and “;” denotes vertical concatenation. Therefore, the LWE sample $\mathbf{c}^T$ can be seen as a commitment to the location of the secret of identity id . Furthermore, the last component of the authentication path defined by $\mathbf{A}_{\text{id}}$ is the secret key $\mathbf{x}_{\text{id}}$ of pk_{id} , while the other components are publicly computable. Thus, the structure of the Merkle tree enables the computation of a membership witness wit_{id} , essentially the authentication path for $\mathbf{y}_{\text{id}}$. Let $\mathbf{u}_{\text{id}_{[0:i]}} := -\mathbf{G}^{-1}(\mathbf{y}_{\text{id}_{[0:i]}\parallel 0}; \mathbf{y}_{\text{id}_{[0:i]}\parallel 1})$ denote the negative of the binary decomposition of $\mathbf{y}_{\text{id}_{[0:i]}\parallel 0}$ concatenated with $\mathbf{y}_{\text{id}_{[0:i]}\parallel 1}$ and let $\text{wit}_{\text{id}} = (\mathbf{u}_{\text{id}_{[0:i]}})_{i=0}^{\ell-1}$ denote their vertical concatenation. Then, $\mathbf{A}_{\text{id}} \cdot (\text{wit}_{\text{id}}; \mathbf{x}_{\text{id}}) = \mathbf{v} \bmod q$ and wit_{id} is low-norm. Thus, anyone in possession of wit_{id} and the secret key $\mathbf{x}_{\text{id}}$ can decrypt the ciphertext $\text{ctxt} = (\mathbf{c}, d)$ in the following way:

$$\mu = \left\lfloor \frac{p}{q} \cdot \left(d - \left\langle \mathbf{c}, \begin{pmatrix} \text{wit}_{\text{id}} \\ \mathbf{x}_{\text{id}} \end{pmatrix} \right\rangle \bmod q \right) \right\rfloor \bmod p.$$

2.2. LE to RBE Transformation

As discussed before, an RBE can be thought of as an LE with the additional efficiency requirement that any registered user only needs to update their membership witness at most $\tilde{O}(\log N)$ times. We give a brief overview of the LE-to-RBE transformation given by DKLLMR [8].

This transformation to RBE employs B LE instances simultaneously for a maximum number of users $N = 2^B$. The state st and reg of the resulting RBE are concatenations of each instances state and registry, i.e. $\text{st} = (\text{st}_1, \dots, \text{st}_B)$ and $\text{reg} = (\text{reg}_1, \dots, \text{reg}_B)$. When a new user joins the system, the public key is added to an empty registry. If two registries contain the same number of registered users, iteratively proceed to merge the registries of these LE instances, leaving one of them in its initial, empty state and the other one with twice as many users. Thus, any identity's public key may only be part of $\log N$ -many merges of two LE instances, which is the only action that requires an update of its witness membership wit_{id} .

To encrypt a message msg the encryptor simply encrypts with respect to all B LE instances. Therefore, the ciphertext size grows *multiplicatively* by the size of B as $\text{ctxt} = (\text{ctxt}_1, \dots, \text{ctxt}_B)$, where each $\text{ctxt}_t \leftarrow \text{Enc}(\text{pp}, \text{st}_t, \text{id}, \text{msg})$ for all $t \in [B]$.

2.3. Our Solution: BLE

Our key observation is that the B parallel LE instances in an RBE are not unrelated. They are all encrypting to the *same* identity id and can share significant structural components. We introduce *Batched Laconic Encryption* (BLE), which enables encrypting to a batch of B instances simultaneously with *additive* rather than multiplicative overhead.

Observe that the DKLLMR LE ciphertext consists of two parts: (1) a large vector $\mathbf{c}^T$ determining the path in the Merkle tree, and (2) a single scalar d . As the vector $\mathbf{c}^T$ determines the same path to identity id in the Merkle tree of each LE instance, our BLE construction generates and wraps it only once by sharing the randomness vector $\mathbf{r}^T$ across all B LE instances. Thus, only the number of scalars $d_1, \dots, d_B$ grows multiplicatively with B . This reduces the ciphertext size from $B \cdot (|\mathbf{c}| + |d|)$ to $|\mathbf{c}| + B \cdot |d|$.

2.4. Other Optimisations

Beyond the architectural shift to BLE, we further employ several optimisations to reduce the ciphertext size.

- *Approximate Gadget*. Rather than using an “exact” gadget matrix $\mathbf{G}$ and performing exact decomposition, we utilise an *approximate* gadget matrix. This allows us to drop low-order bits during decomposition, significantly reducing the dimension of $\mathbf{A}_{\text{id}}$.
- *Computational Reductions*. The security proof in [8] relies on a statistical argument for indistinguishability of the public keys, namely, the leftover hash lemma, which requires large parameters. Instead, we reduce to the LWE assumption, which is required anyway, resulting in smaller parameters.
- *Optimal Tree-Arity*. We observe that the ciphertext size scales linearly with $k/\log k$, where k denotes the arity of the Merkle tree. We minimise this by choosing $k = 3$, relying on ternary trees.

We discuss these and a few more optimisations in more detail in Sections 5 and 6.

3. Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter. For $n \in \mathbb{N}$, write $[n] = \{0, 1, \dots, n-1\}$. For a finite set S we write $a \leftarrow_{\$} S$ for sampling uniformly from the set S . For a modulus $q \in \mathbb{N}$, we write $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$, with elements represented by $(-q/2, q/2] \cap \mathbb{Z}$. For dimension k implicit from the context, we write $\mathbf{i}_j^T \in \{0, 1\}^k$ for the (horizontal) j -th standard unit vector of dimension k . We write ϵ for the empty (k -ary) string. For a (k -ary) string str of length ℓ and $0 \leq j \leq \ell$, we write $\text{str}[j]$ for the length- j prefix str . For any pair of experiments Exp-X^0 and Exp-X^1 , we define the advantage function $\text{Adv-X}(\lambda) := |\Pr[\text{Exp-X}^0(1^\lambda) = 1] - \Pr[\text{Exp-X}^1(1^\lambda) = 1]|$.

3.1. Algebraic Number Theory and Lattices

Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be the ring of integers of the cyclotomic field $\mathbb{Q}(\zeta)$ of conductor $\mathfrak{f}$ and degree $\varphi = \varphi(\mathfrak{f})$. For $q \in \mathbb{N}$, we write $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. We represent elements $a = \sum_{i=0}^{\varphi-1} a_i \zeta^i$ in $\mathcal{R}$ and $\mathcal{R}_q$ as $\mathbb{Z}$ - and $\mathbb{Z}_q$ -linear combinations of the standard power basis $(1, \zeta, \dots, \zeta^{\varphi-1})$, respectively. The uniform distribution over $\mathcal{R}_q$, denoted $\mathcal{U}(\mathcal{R}_q)$, is induced by sampling each coefficient uniformly over $(-q/2, q/2] \cap \mathbb{Z}$. For an element $a = \sum_{i=0}^{\varphi-1} a_i \zeta^i \in \mathcal{R}$, its (coefficient) ℓ_p -norm is defined as the ℓ_p norm of $(a_i)_{i=0}^{\varphi-1}$. For a vector $\mathbf{x} = (x_i)_{i=0}^{n-1} \in \mathcal{R}^n$, its (coefficient) ℓ_p -norm is defined as the ℓ_p norm of $(x_i)_{i=0}^{n-1}$. For a matrix $\mathbf{M} \in \mathcal{R}^{n \times m}$, we write $\|\mathbf{M}\|_\infty := \|\text{vec}(\mathbf{M})\|_\infty$ where vec denotes vectorisation. The ℓ_p -expansion factor of $\mathcal{R}$ is $\gamma_p := \max_{a, b \in \mathcal{R} \setminus \{0\}} \frac{\|a \cdot b\|_p}{\|a\|_p \cdot \|b\|_p}$. For $\mathfrak{f}$ a power of 2, we have $\gamma_2 \leq \sqrt{\varphi}$.

Exp-LWE _{par,A} ^b (1 ^λ)	Exp-LLWE _{par,A} ^b (1 ^λ)
$\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n \times m}$	$\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n \times m}$
$\mathbf{r} \leftarrow_{\$} \chi_r^n; \mathbf{e} \leftarrow_{\$} \chi_e^m$	$\mathbf{r} \leftarrow_{\$} \chi_r^m; \mathbf{e} \leftarrow_{\$} \chi_e^m; \bar{\mathbf{e}} \leftarrow_{\$} \bar{\chi}_e^Q$
if $b = 0$ then $\mathbf{b}^T := \mathbf{r}^T \cdot \mathbf{A} + \mathbf{e}^T \bmod q$	if $b = 0$ then $\mathbf{b}^T := \mathbf{r}^T \cdot \mathbf{A} + \mathbf{e}^T \bmod q$
if $b = 1$ then $\mathbf{b} \leftarrow_{\$} \mathcal{R}_q^m$	if $b = 1$ then $\mathbf{b} \leftarrow_{\$} \mathcal{R}_q^m$
return $\mathcal{A}(\mathbf{A}, \mathbf{b})$	$(\mathbf{L}_r, \mathbf{L}_e) \leftarrow \mathcal{A}(\mathbf{A}); \text{assert } (\mathbf{L}_r, \mathbf{L}_e) \in \mathcal{L}$
	$\mathbf{l}^T := (\mathbf{r}^T, \mathbf{e}^T) \begin{pmatrix} \mathbf{L}_r \\ \mathbf{L}_e \end{pmatrix} + \bar{\mathbf{e}}^T;$
	return $\mathcal{A}(\mathbf{b}, \mathbf{l})$

Figure 2: LWE and Leaky LWE experiments

Proposition 3.1 ([25, Proposition 2]). *If $\mathfrak{f}$ is a prime-power then, $\gamma_\infty \leq 2\varphi$. If $\mathfrak{f}$ is a power of 2 then $\gamma_\infty \leq \varphi$.*

A lattice $\Lambda \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$. The module $\mathcal{R}^m$ can be viewed as a lattice in $\mathbb{R}^{\varphi m}$ through its coefficient embedding. For any Gaussian parameter $s > 0$ and centre $\mathbf{c} \in \mathbb{R}^m$, define the Gaussian function $\forall \mathbf{x} \in \mathbb{R}^m : \rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / s^2)$. The centre $\mathbf{c}$ is taken to be $\mathbf{0}$ when omitted. For any lattice $\Lambda \subset \mathbb{R}^m$, $s > 0$, $\mathbf{c} \in \mathbb{R}^m$, the *discrete Gaussian distribution* over Λ is defined as $\forall \mathbf{x} \in \Lambda : D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$, where $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$ is the normalisation factor. For $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ of Λ is the smallest s such that $\rho_{1/s}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$, where Λ^* is the dual lattice of Λ . The discrete Gaussian distribution over $\mathcal{R}$ with parameter s , denoted by $D_{\mathcal{R},s}$, is induced by sampling φ independent samples $x_i \leftarrow_{\$} D_{\mathbb{Z},s}$ and outputting $x = \sum_{i \in [\varphi]} x_i \cdot \zeta^i$.

Lemma 3.2 ([26, Lemma 3.3]). *For any rank- n lattice Λ and positive real $\epsilon > 0$, $\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n(\Lambda)$ where $\lambda_n(\Lambda)$ is the n -th successive minimum of Λ .*

Lemma 3.3 (Adapted from [27, Lemma 2.6]). *Let $\Lambda \subset \mathbb{R}^n$ be a lattice, and $s > 0$. It holds that $\Pr[\|\mathbf{x}\|_2 > s\sqrt{n} \mid \mathbf{x} \leftarrow_{\$} D_{\Lambda,s}] \leq 2^{-n}$.*

Corollary 3.4 ([28, Corollary 4.7]). *Let Λ be a full-rank n -dimensional lattice and $s, \bar{s} \geq \sqrt{2} \cdot \eta_\epsilon(\Lambda)$ for some $\epsilon < \frac{1}{4}$. Then $D_{\Lambda,s} + D_{\Lambda,\bar{s}}$ is within 16ϵ statistical distance of $D_{\Lambda,\bar{s}}$, where $\bar{s}^2 = s^2 + \bar{s}^2$.*

Proof. The claim follows almost immediately from [28, Corollary 4.7], except for the statistical distance bound. The latter follows easily from [28, Theorem 4.6], using that max-log distance at most $K = \log(\frac{1+\epsilon'}{1-\epsilon'})$, where $1 + \epsilon' = (1 + \epsilon)^2$, implies ∞ -Renyi divergence at most K , and thus statistical distance at most $2^K - 1$. Simplifying, we get $2^K - 1 = \frac{2\epsilon'}{1-\epsilon'} \leq 4\epsilon' \leq 16\epsilon$, where we used $1 + 2x \leq (1 + x)^2 \leq 1 + 4x$ for $x \in [0, 1/4]$. $\square$

3.2. Computational Assumptions

We recall the standard (module) Learning with Errors (LWE) assumption [29], [30] and (a special case of) the Leaky LWE assumption [31].

Definition 3.5 (LWE). Let $\text{par}(\lambda) = (\mathcal{R}, n, m, q, \chi_r, \chi_e)$. The LWE_{par} assumption states that for any PPT adversary $\mathcal{A}$, the advantage $\text{Adv-LWE}_{\text{par},\mathcal{A}}^{\text{par}} :=$

$$|\Pr[\text{Exp-LWE}_{\text{par},\mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp-LWE}_{\text{par},\mathcal{A}}^1(1^\lambda) = 1]|$$

is negligible in λ , where $\text{Exp-LWE}_{\text{par},\mathcal{A}}^b$ given in Figure 2.

Definition 3.6 (Leaky LWE [31]). Let $\text{par}(\lambda) = (\mathcal{R}, n, m, q, Q, \chi_r, \chi_e, \bar{\chi}_e, \mathcal{L})$ where $\mathcal{L} \subset \mathcal{R}^{m \times Q}$ is an efficiently decidable set. The (decision) LLWE_{par} assumption states that for any PPT adversary $\mathcal{A}$, $\text{Adv-LLWE}_{\text{par},\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$ for $\text{Exp-LLWE}_{\text{par},\mathcal{A}}^b$ given in Figure 2.

Lai, Swarnakar and Woo [31] provided a reduction from LWE to Leaky LWE, which is stated with respect to the spectral norm of the leakage matrix with respect to any fixed choice of embedding. We state an implied reduction in terms of the ℓ_∞ -norm over the coefficient embedding.

For a matrix $\mathbf{L}$ over $\mathcal{R}^{h \times k}$, let $\Psi(\mathbf{L}) \in \mathbb{Z}^{\varphi h \times \varphi k}$ be the matrix representing the $\mathbb{Z}$ -linear map sending the coefficient-embedding of $\mathbf{x} \in \mathcal{R}^k$ to the coefficient-embedding of $\mathbf{L}\mathbf{x} \in \mathcal{R}^h$. Suppose $\mathcal{R}$ is a power-of-2 cyclotomic ring and $\mathbf{L}$ has at most w non-zero entries, denoted $|\mathbf{L}| \leq w$, each with ℓ_∞ -norm at most β , then $\Psi(\mathbf{L})$ has $w\varphi^2$ non-zero entries each with absolute value at most β . The Frobenius norm, and hence the spectral norm, of $\Psi(\mathbf{L})$ is then at most $\sqrt{w}\varphi\beta$.

Lemma 3.7 (Adapted from [31, Theorem 3, Example 1]). Let $\text{lwepar}(\lambda) = (\mathcal{R}, n, m, q, \mathcal{D}_{\mathcal{R}, \sigma^*}, \mathcal{D}_{\mathcal{R}, \sigma^*})$ and $\text{llwepar}(\lambda) = (\mathcal{R}, n, m, q, Q, \mathcal{D}_{\mathcal{R}, \sigma}, \mathcal{D}_{\mathcal{R}, \sigma}, \mathcal{D}_{\mathcal{R}, \bar{\sigma}}, \mathcal{L})$ be *LWE* and *Leaky LWE* parameters, respectively, for some Gaussian parameters $\sigma, \bar{\sigma}, \sigma^* > 0$. Assume the following properties:

- $\mathcal{R}$ is a degree- φ cyclotomic ring with a power-of-2 conductor viewed as a lattice $\mathbb{Z}^\varphi$ via the coefficient embedding with respect to the standard power basis.
- $\sigma^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathbb{Z}^{\varphi m})$ for some $\epsilon > 0$.
- $\mathcal{L} = \{\mathbf{L} \in \mathcal{R}^{m \times Q} : \|\mathbf{L}\| \leq w \wedge \|\mathbf{L}\|_\infty \leq \beta\}$ where

$$\frac{w\varphi^2\beta^2}{\bar{\sigma}^2} + \frac{1}{\sigma^2} \leq \frac{1}{2\eta_\epsilon(\mathbb{Z}^{\varphi(n+m)})^2 + (\sigma^*)^2}.$$

It holds that

$$\text{Adv-LWE}_{\text{lwepar}, \mathcal{A}}(\lambda) \geq \text{Adv-LLWE}_{\text{llwepar}, \mathcal{A}}(\lambda) - 4\epsilon/(1 - \epsilon).$$

3.3. Approximate Gadget Decomposition

Fix a module rank $n \in \mathbb{N}$, a base $b \in \mathbb{N}$ for radix decomposition and a modulus $q \in \mathbb{N}$. Define the gadget vector $\mathbf{g}^T = (1, b, \dots, b^{\lceil \log_b q \rceil - 1}) \in \mathcal{R}_q^{\lceil \log_b q \rceil}$ and the gadget matrix $\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}^T \in \mathcal{R}_q^{n \times n \lceil \log_b q \rceil}$ (cf. [27]). For any $\mathbf{y} \in \mathcal{R}_q^n$, write $\mathbf{x} = \mathbf{G}^{-1}(\mathbf{y})$ for the coefficient-wise (w.r.t. the power basis $(1, \zeta, \dots, \zeta^{\varphi-1})$) b -ary decomposition of $\mathbf{y}$, also known as the gadget decomposition of $\mathbf{y}$. We have $\|\mathbf{x}\|_\infty \leq b/2$ and $\mathbf{G}\mathbf{x} = \mathbf{y}$.

Following [32], some entries of $\mathbf{x} = \mathbf{G}^{-1}(\mathbf{y})$ can be dropped to yield an *approximate* gadget decomposition of $\mathbf{y}$. Concretely, fix $h \in \mathbb{N}$ with $h < \lceil \log_b q \rceil$. Define the approximate gadget vector $\tilde{\mathbf{g}}^T = (b^h, \dots, b^{\lceil \log_b q \rceil - 1}) \in \mathcal{R}_q^{\lceil \log_b q \rceil - h}$ and the approximate gadget matrix $\tilde{\mathbf{G}} := \mathbf{I}_n \otimes \tilde{\mathbf{g}}^T \in \mathcal{R}_q^{n \times n(\lceil \log_b q \rceil - h)}$. For any $\mathbf{y} \in \mathcal{R}_q^n$, the approximate gadget decomposition $\tilde{\mathbf{x}} = \tilde{\mathbf{G}}^{-1}(\mathbf{y})$ is obtained by dropping the first hn entries from $\mathbf{G}^{-1}(\mathbf{y})$. We have $\|\tilde{\mathbf{x}}\|_\infty \leq b/2$ and $\|\tilde{\mathbf{G}}\tilde{\mathbf{x}} - \mathbf{y} \pmod{q}\|_\infty < b^h/2$.

4. Batched Laconic Encryption

We define a generalisation of laconic encryption (LE) [8] called batched laconic encryption (BLE). A BLE allows anyone to encrypt a message to an identity registered in a set S of instances of LE. Without any efficiency constraints, it is trivial to realise a BLE using $|S|$ parallel instances of LE. We are therefore interested in BLE schemes satisfying a non-trivial efficiency requirement (Definition 4.6) where the ciphertext size grows sublinear in $|S|$ (relative to some base LE scheme).

In more detail, recall that an LE registry reg maintains a set of at most N registered (id, pk) tuples with a short digest stored in the system state st of size sublinear in N . Anyone, with access to st (but not necessarily the full registry reg), can encrypt a message to any registered user id , who can use the secret key sk associated to pk and some publicly computable membership witness wit to decrypt. A BLE registry reg maintains a set of sub-registries indexed by $S \subseteq [B]$ each storing (id, pk) tuples, which can be added, updated or removed in any order throughout the lifetime of the system. The size of state st could depend on $|S|$ but must be sublinear in N . Anyone with access to st can encrypt a set of messages $(S, (\text{msg}_t)_{t \in S})$ to an identity id . For each $t \in S$, if (id, pk) is registered in the t -th sub-registry, then the holder of the corresponding secret key sk and membership witness wit can decrypt the message msg_t .

4.1. Formal Definitions

Definition 4.1 (BLE). A batched laconic encryption (BLE) scheme for an identity space $\mathcal{ID} \subseteq [k]^{\leq \hat{\ell}}$ for some arity $k \in \mathbb{N}$ and maximum identity length $\hat{\ell}$, maximal batch size $B \in \mathbb{N}$, and message space $\mathcal{M}$ consists of PPT algorithms (Setup, KGen, Enc, Dec, Upd, WGen) with the following syntax:

$(\text{pp}, \text{st}, \text{reg}) \leftarrow \text{Setup}(1^\lambda)$ generates (1) the public parameters pp which implicitly defines a family of ciphertext spaces $\mathcal{C}_{S, \ell}$ for $S \subseteq [B]$ and $\ell \in \mathbb{N}$ and is fixed throughout the lifetime of the system, (2) a state st , and (3) a registry reg which implicitly contains a set of sub-registries indexed by an index set $S \subseteq [B]$, each storing (id, pk) tuples.

$(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{pp})$ generates a pair of public and secret keys (pk, sk) .

$\text{ctxt} \leftarrow \text{Enc}(\text{pp}, \text{st}, S, \text{id}, (\text{msg}_t)_{t \in S})$ encrypts a set of messages $(\text{msg}_t)_{t \in S}$ with respect to an index set $S \subseteq [B]$ and an identity $\text{id} \in \mathcal{ID}$ into a ciphertext $\text{ctxt} \in \mathcal{C}_{S, \ell}$.

$\text{msg} \leftarrow \text{Dec}(\text{sk}, \text{wit}, \text{ctxt})$ decrypts a ciphertext ctxt using a secret key sk and a membership witness wit and outputs a message $\text{msg} \in \mathcal{M}$.

$\text{st}' \leftarrow \text{Upd}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id}, \text{pk})$ with (read and write) random access to the registry reg *deterministically* registers the tuple (id, pk) in the t -th sub-registry if $\text{pk} \neq \perp$, or removes id from the t -th sub-registry if $\text{pk} = \perp$.

$\text{wit} \leftarrow \text{WGen}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id})$ *deterministically* generates a membership witness for (id, pk) in the t -th sub-registry.

$\text{Correctness}_{\text{BLE}, \mathcal{A}}(1^\lambda)$ $(\text{pp}, \text{st}, \text{reg}) \leftarrow \text{Setup}(1^\lambda)$ $(S^*, \text{id}^*, (\text{msg}_t^*)_{t \in S^*}) \leftarrow \mathcal{A}^{\text{KGenO}, \text{UpdO}}(\text{pp}, \text{st}, \text{reg})$ $\text{ctxt}^* \leftarrow \text{Enc}(\text{pp}, \text{st}, S, \text{id}, (\text{msg}_t^*)_{t \in S^*})$ for $t \in S^* : \text{SK}[\text{REG}[t, \text{id}^*]] \neq \perp$ do $\text{sk}_t \leftarrow \text{SK}[\text{REG}[t, \text{id}^*]]$ $\text{wit}_t \leftarrow \text{WGen}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id}^*)$ assert $\text{Dec}(\text{sk}_t, \text{wit}_t, \text{ctxt}^*) = \text{msg}_t^*$ return 1	
$\text{KGenO}()$ $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{pp})$ $\text{SK}[\text{pk}] := \text{sk}$ return pk	$\text{UpdO}(t, \text{id}, \text{pk})$ $\text{st} \leftarrow \text{Upd}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id}, \text{pk})$ $\text{REG}[t, \text{id}] = \text{pk}$ return (st, reg)
$\text{Exp-IND\$-CPA}_{\text{BLE}, \mathcal{A}}^b(1^\lambda)$ $(\text{pp}, \text{st}, \text{reg}) \leftarrow \text{Setup}(1^\lambda)$ $(S^*, \text{id}^*, (\text{msg}_t^*)_{t \in S^*}) \leftarrow \mathcal{A}^{\text{KGenO}, \text{UpdO}}(\text{pp}, \text{st}, \text{reg})$ for $t \in S^*$ assert $\text{REG}[t, \text{id}^*] = \perp \vee \text{SK}[\text{REG}[t, \text{id}^*]] \neq \perp$ if $b = 0$ then $\text{ctxt}^* \leftarrow \text{Enc}(\text{pp}, \text{st}, S^*, \text{id}^*, (\text{msg}_t^*)_{t \in S^*})$ else $\text{ctxt}^* \leftarrow \mathcal{C}_{S^*, \text{id}^* }$ $b' \leftarrow \mathcal{A}^{\text{KGenO}, \text{UpdO}}(\text{ctxt}^*)$ return b'	

Figure 3: BLE experiments. Each experiment implicitly initialises an empty REG for storing registered pk and an empty SK for storing honestly generated sk.

Definition 4.2 (LE). A laconic encryption (LE) scheme for an identity space $\mathcal{ID} \subseteq [k]^{\leq \ell}$, an arity $k \in \mathbb{N}$, a maximum identity length ℓ , and a message space $\mathcal{M}$, is a batched laconic encryption (BLE) scheme restricted to batch size $B = 1$.

Definition 4.3 (RBE). A registration-based encryption (RBE) scheme for an identity space $\mathcal{ID}$, a message space $\mathcal{M}$, and a maximum number of users N is a laconic encryption scheme that satisfies the following additional efficiency requirement: **Update Efficiency:** For any identity $\text{id} \in \mathcal{ID}$, the total number of updates to the membership witness wit for id is at most $\tilde{O}(\log N)$ over the lifespan of the system.

Definition 4.4 (Correctness). A BLE is said to be ϵ -correct if for any (unbounded) algorithm $\mathcal{A}$ it holds that

$$\Pr[\text{Correctness}_{\text{BLE}, \mathcal{A}}(1^\lambda) = 1] \geq 1 - \epsilon$$

where $\text{Correctness}_{\text{BLE}, \mathcal{A}}$ is defined in Figure 3.

Definition 4.5 (Batching Overhead). Let $r > 0$ and $|\text{ctxt}_{|S|}|$ define the size of a BLE ciphertext for any batch of identities $S \subseteq [B]$. A BLE scheme BLE is said to be r -efficient if

$$\lim_{|S| \rightarrow \infty} \frac{|\text{ctxt}_{|S|}|}{|S| \cdot |\text{ctxt}_1|} \leq r$$

and $|\text{ctxt}_{|S|}| \leq |\text{ctxt}_{|S'|}|$ whenever $S \subseteq S'$.

Definition 4.6 (Non-trivial Efficiency). A BLE scheme is said to be non-trivially efficient if it is r -efficient with $r < 1$.

Remark 4.7. The naive construction of BLE from multi-instance laconic encryption, as described in the RBE resulting from the LE-to-RBE transform in Section 2.2, does not satisfy the non-trivial efficiency requirement. The batching overhead is $\lim_{|S| \rightarrow \infty} \frac{|S| \cdot |\text{ctxt}_{|S|}^{\text{LE}}|}{|S| \cdot |\text{ctxt}_1^{\text{LE}}|} = 1$.

Definition 4.8 (IND\\$-CPA Security). A BLE is said to be pseudorandom under chosen plaintext attack (IND\\$-CPA) if for any PPT (stateful) adversary $\mathcal{A}$ the advantage $\text{Adv-IND\$-CPA}_{\text{BLE}, \mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$ for $\text{Exp-IND\$-CPA}_{\text{BLE}, \mathcal{A}}^b$ defined in Figure 3.

Remark 4.9. The condition $\text{SK}[\text{REG}[t, \text{id}^*]] \neq \perp$ in the pseudorandomness experiment in Figure 3 is included to ensure that the corresponding public key is honestly generated. For simplicity, our security definition assumes that all elements in the challenge set are honestly generated, and we prove that the challenge ciphertext is pseudorandom in its entirety. Indeed, this is the case for the BLE-to-RBE transformation. A more fine-grained security notion could alternatively allow $\text{SK}[\text{REG}[t, \text{id}^*]] = \perp$ for some t (i.e., allowing some maliciously generated public keys), while requiring pseudorandomness for the ciphertext components associated with those t such that $\text{SK}[\text{REG}[t, \text{id}^*]] \neq \perp$.

As in the definition of LE [8], Definitions 4.4 and 4.8 model correctness and security, respectively, assuming an honest-but-curious key curator that is expected to perform Upd and WGen honestly. Since the WGen and Upd algorithms are deterministic, all computation performed by the key curator can be repeated and verified publicly.

Although we do not formalise this specific efficiency requirement for LE, BLE, and RBE, we keep in mind that these notions are only interesting when the size of the system state st remains sublinear in N . While Definition 4.8 defines the

IND\$-CPA security of a BLE scheme, a definition of the chosen-ciphertext attack variant (IND\$-CCA) can be obtained by equipping the adversary with an additional decryption oracle for keys registered in Honest. It is easy to see that an IND\$-CPA-secure BLE scheme can be upgraded to an IND\$-CCA-secure one using generic transformations such as that of Fujisaki and Okamoto [33].

4.2. Transforming BLE to RBE

A BLE generically implies an RBE using a transform similar to that from LE to RBE [8].

Theorem 4.10. *If there exists a BLE scheme BLE for an identity space $\mathcal{ID}$ and message space $\mathcal{M}$ with maximum batch size $B \in \mathbb{N}$, then there exists an RBE scheme for the same spaces supporting a maximum of $N = 2^B$ users.*

Proof. (Sketch.) We construct an RBE for at most $N = 2^B$ users which is simply a BLE with batch size B with a particular way of managing the underlying data structure. In brief, an invariant is maintained so that the RBE identities are stored in BLE sub-registries whose sizes are progressively larger powers of two. A new RBE identity is inserted in the smallest (possibly empty) BLE sub-registry. If two sub-registries are of the same size, they are merged into a larger sub-registry so that all sub-registries are of size distinct powers of 2. Every merge operation results in one of the two sub-registries containing twice the number of identities as before and one in an empty state. Clearly, this meets the update efficiency requirement of RBE that, as more users are registered, the membership witness wit of each registered user is updated at most $\mathcal{O}(\log N)$ times, where N denotes the maximum number of registered users. $\square$

Compared to the LE-to-RBE transform [8], this transform programs the B sub-registries of BLE to take on the role of the registries of B LE instances.

5. BLE Construction

We construct a BLE scheme and prove its IND\$-CPA security from the LWE assumption. Our BLE is an extension of the LE of [8]. We first recall the latter and then describe how we extend it to a BLE.

LE of DKLLMR. The LE construction of [8] recalled in Section 2.1 utilises a cyclotomic ring $\mathcal{R} = \mathbb{Z}[\zeta]$, is parametrised by n, m, ℓ, p, q where $m = n \lceil \log q \rceil$ and implicitly assumes that identities are strings in $\{0, 1\}^\ell$. Its correctness builds on the following fact:

$$\begin{pmatrix} \mathbf{A}_0 \parallel \mathbf{A}_1 \\ \mathbf{i}_{\text{id}_i}^\top \otimes \mathbf{G} \end{pmatrix} \cdot \underbrace{\begin{pmatrix} -\mathbf{G}^{-1} \left(\mathbf{y}_{\text{id}_{[0:i]}} \parallel 0 \right) \\ -\mathbf{G}^{-1} \left(\mathbf{y}_{\text{id}_{[0:i]}} \parallel 1 \right) \end{pmatrix}}_{=\mathbf{u}_{\text{id}_{[0:i]}}} = \begin{pmatrix} \mathbf{y}_{\text{id}_{[0:i]}} \\ -\mathbf{y}_{\text{id}_{[0:i+1]}} \end{pmatrix}$$

where $\mathbf{y}_{\text{id}_{[0:i]}} = H \left(\mathbf{y}_{\text{id}_{[0:i]}} \parallel 0, \mathbf{y}_{\text{id}_{[0:i]}} \parallel 1 \right)$ and $\mathbf{u}_{\text{id}_{[0:i]}}$ denotes the corresponding part in the membership witness wit_{id} . By chaining these blocks with overlap (as in $\mathbf{A}_{\text{id}}$), the positive and negative values of $\mathbf{y}_{\text{id}_{[0:i]}}$ cancel each other out, implying

$$\mathbf{A}_{\text{id}} \cdot \begin{pmatrix} \text{wit}_{\text{id}} \\ \mathbf{x}_{\text{id}} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_\epsilon^\top & \mathbf{0}^\top & \dots & \mathbf{0}^\top \end{pmatrix}^\top = \mathbf{v} \bmod q,$$

as the public key $\mathbf{y}_{\text{id}} = \mathbf{B} \cdot \mathbf{x}_{\text{id}}$ completes this chain. This equation ensures that decryption based on dual-Regev [3] with respect to the tuple $(\mathbf{A}_{\text{id}}, \mathbf{v})$ works:

$$\begin{aligned} & \left\lfloor \frac{p}{q} \cdot (d - \langle \mathbf{c}, \text{wit}_{\text{id}} \rangle \bmod q) \right\rfloor \bmod p \\ &= \left\lfloor p/q \cdot (\mathbf{r}^\top \mathbf{v} + \tilde{e} + \lfloor q/p \rfloor \cdot \mu \right. \\ & \quad \left. - (\mathbf{r}^\top \mathbf{A}_{\text{id}} + \mathbf{e}^\top) \cdot \text{wit}_{\text{id}} \parallel \mathbf{x}_{\text{id}} \bmod q) \right\rfloor \bmod p \\ &= \left\lfloor \frac{p}{q} \cdot \left(\tilde{e} + \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu - \left\langle \mathbf{e}, \begin{pmatrix} \text{wit}_{\text{id}} \\ \mathbf{x}_{\text{id}} \end{pmatrix} \right\rangle \bmod q \right) \right\rfloor \bmod p, \end{aligned}$$

which equals μ if the error terms are small enough.

Our BLE. Suppose the generic transformation described in [8] and recalled in Theorem 4.10 is applied to turn the above LE into an RBE. We observe that an RBE ciphertext destined to id consists of $B := \log N$ instances of LWE samples w.r.t. the matrix $\mathbf{A}_{\text{id}}$. This is a major source of inefficiency in the RBE construction. Our idea of eliminating this inefficiency is to reuse a single instance of LWE samples w.r.t. the matrix $\mathbf{A}_{\text{id}}$ for all B underlying LE ciphertexts. Formally, we describe this in the language of BLE.

In a nutshell, our BLE scheme contains dynamically maintained instances of the LE scheme of DKLLMR, i.e. the BLE registry reg and state st consist of multiple instances of the LE registry and state, respectively. Let $S \subseteq [B]$ be an index set⁴ indexing a subset of LE instances and let $\mathbf{Y}_{S,\epsilon}$ be the matrix whose t -th column is given by the t -th state $\mathbf{y}_{t,\epsilon}$. Correspondingly, let $\mathbf{V}_S \in \mathcal{R}_q^{(\ell+1)n \times S}$ be the matrix whose top n rows are given by $\mathbf{Y}_{S,\epsilon}$ and the rest by zeros. To encrypt a set of messages $\mathbf{m} \in \mathcal{R}_p^S$, the encryptor samples a uniformly random vector $\mathbf{r} \leftarrow_{\S} \mathcal{R}_q^{(\ell+1)n}$ and short errors $\mathbf{e} \leftarrow_{\S} \mathcal{R}^{(2\ell+1)m}$ and $\tilde{\mathbf{e}} \leftarrow_{\S} \mathcal{R}^S$ and computes

$$\begin{aligned} \mathbf{c}^T &:= \mathbf{r}^T \mathbf{A}_{\text{id}} + \mathbf{e}^T \bmod q, \\ \mathbf{d}^T &:= \mathbf{r}^T \mathbf{V}_S + \tilde{\mathbf{e}}^T + \left\lfloor \frac{q}{p} \right\rfloor \cdot \mathbf{m}^T \bmod q. \end{aligned}$$

If user id is registered at some instance $t \in S$ of the underlying LEs, they can decrypt the ciphertext and obtain m_t by running LE decryption on $(\mathbf{c}, \mathbf{d}_t)$, which is essentially a dual-Regev ciphertext decryption w.r.t. public key $\mathbf{A}_{\text{id}}$.

Further Optimisations. Our formal construction further incorporates additional efficiency optimisations.

- (1) DKLLMR uses a binary Merkle tree. Suppose it is replaced by a k -ary tree, we notice that the ciphertext size scales linearly in $k/\log k$, which is minimised when $k = 3$ (for $k \in \mathbb{N}$).
- (2) Instead of performing decomposition with base 2, we could decompose labels with a much larger base b , e.g. $b \approx q^{1/3}$. This reduces the number of columns $km_{\mathbf{A}}$ of $\mathbf{A}$ from $kn \log_2(q)$ to $kn \log_b(q)$.
- (3) Instead of using the exact decomposition operator $\mathbf{G}^{-1}$, we replace it with an approximate decomposition operator $\tilde{\mathbf{G}}^{-1}$ (and the respective gadget matrices), dropping the least significant digits $b, \dots, b^{h-h}$, where b and h determine the approximation quality. The (cumulative) effect of this change is that the dimension of the matrices $\mathbf{A}_0, \dots, \mathbf{A}_{k-1}, \mathbf{B}$ in the public parameters is further reduced to $kn(\log_b(q) - h)$. We additionally change the distribution of the encryption randomness $\mathbf{r}$ from uniform to a low-norm distribution, as this is needed in our security proof. This modification is a standard practice known as “normal-form LWE” in the literature.
- (4) We change the format of the key from $\mathbf{y} = \mathbf{B}\mathbf{x} \bmod q$ to $\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{z} \bmod q$, where $\mathbf{x}, \mathbf{z}$ are short vectors, and we let $\mathbf{B}$ have much fewer columns $m_{\mathbf{B}}$ to shrink the ciphertext size. In the security proof, the dimension $m_{\mathbf{B}}$ is so small that we could no longer apply the leftover hash lemma to argue that e.g. $\mathbf{y} = \mathbf{B}\mathbf{x} \bmod q$ is statistically close to uniformly random. Instead, we rely on LWE to argue that $\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{z} \bmod q$ is pseudorandom.
- (5) The security analysis of DKLLMR is based on the Error-Leakage LWE assumption [8] which is shown to be implied by LWE in the same work. In our security proof, we have to argue about the security of LWE in presence of both secret- and error-leakages, due to the use of the approximate gadget. We therefore instead rely on the Leaky LWE assumption [31], which generalises the Error-Leakage LWE assumption, that captures both secret- and error-leakages and is shown [31] to be implied by LWE. We further obtain a slight efficiency gain since the reduction in [31] is tighter than that in [8], even when restricting only to error-leakages.

In addition, we lift the restriction that all identities are strings of the same length, but still insist that all registered identities are prefix-free.

5.1. Formal Description

Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be a cyclotomic ring, $\mathcal{ID} = [k]^{\leq \ell}$ be the identity space, $B \in \mathbb{N}$ be a maximal batch size, $p < q$ be moduli, b, h with $h < \lceil \log_b q \rceil$ be parameters for approximate gadget decomposition, $n, m_{\mathbf{A}} = n(\lceil \log_b q \rceil - h), m_{\mathbf{B}}$ be matrix dimensions, $\chi_x, \chi_z, \chi_r, \chi_e, \tilde{\chi}_e$ be distributions over $\mathcal{R}$, all parametrised by λ . We construct a BLE scheme BLE in Figures 4 and 5.

Remark 5.1. When encrypting to multiple identities, it is possible to reuse the parts of the ciphertexts that correspond to the common prefixes of these identities without compromising security. It is important, however, to be cautious that the encryption randomness not corresponding to common prefixes should not be reused.

5.2. Analysis

We analyse the correctness and the security of BLE.

⁴We will also use S to index dimensions of matrices and vectors.

$\text{Enc}(\text{pp}, \text{st}, S \subseteq [B], \text{id} \in [k]^{\leq \hat{\ell}}, \mathbf{m} \in \mathcal{R}_p^S)$	$\text{Setup}(1^\lambda)$	$\text{KGen}(\text{pp})$
$\ell := \text{id} $ / Length of id / Digests associated to the target set S $\mathbf{Y}_{S, \epsilon} := [\mathbf{y}_{t, \epsilon} = \text{st}[t, \epsilon] : t \in S]$ $\mathbf{r}_j \leftarrow_{\$} \chi_r^n \quad \forall j \in \{0, 1, \dots, \ell\}$ for $j \in \{0, 1, \dots, \ell - 1\}$ do $\mathbf{e}_j \leftarrow_{\$} \chi_e^{k \cdot m_A}$ $\mathbf{c}_j^T := (\mathbf{r}_j^T \quad \mathbf{r}_{j+1}^T) \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} + \mathbf{e}_j^T \bmod q$ $\mathbf{e}_\ell \leftarrow_{\$} \chi_e^{m_B}$ $\mathbf{c}_\ell^T := \mathbf{r}_\ell^T \cdot \mathbf{B} + \mathbf{e}_\ell^T \bmod q$ $\tilde{\mathbf{e}} := (\tilde{e}_t)_{t \in S} \leftarrow_{\$} \tilde{\chi}_e^S$ $\mathbf{d}^T := (d_t)_{t \in S} := \mathbf{r}_0^T \cdot \mathbf{Y}_{S, \epsilon} + \tilde{\mathbf{e}}^T + \left\lfloor \frac{q}{p} \right\rfloor \cdot \mathbf{m}^T \bmod q$ return $\text{ctxt} := (\mathbf{c}_0, \dots, \mathbf{c}_\ell, \mathbf{d})$	$\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n \times k \cdot m_A}; \quad \mathbf{B} \leftarrow_{\$} \mathcal{R}_q^{n \times m_B}$ for $t \in [B]$ do / initiate sub-registries with terminators $\text{term}[t] := \mathbf{y}_{t, \dagger} \leftarrow_{\$} \mathcal{R}_q^n$ $\text{st}[t] := \text{reg}[t, \epsilon] := \text{term}[t]$ $\text{pp} := (\mathbf{A}, \mathbf{B}, \text{term})$ return $(\text{pp}, \text{st}, \text{reg})$	$\mathbf{x} \leftarrow_{\$} \chi_x^{m_B}; \quad \mathbf{z} \leftarrow_{\$} \chi_z^n$ $\mathbf{y} := \mathbf{B} \cdot \mathbf{x} + \mathbf{z} \bmod q$ $\text{pk} := \mathbf{y}; \quad \text{sk} := \mathbf{x}$ return (pk, sk)
	$\text{Dec}(\text{sk} = \mathbf{x}_{t, \text{id}}, \text{wit} = (t, (\mathbf{u}_{t, \text{id}[j]})_{j \in [\ell]}), \text{ctxt})$ $\tilde{\mu} := d_t - \sum_{j \in [\ell]} \mathbf{c}_j^T \cdot \mathbf{u}_{t, \text{id}[j]} - \mathbf{c}_\ell^T \cdot \mathbf{x}_{t, \text{id}} \bmod q$ return $\left\lfloor \frac{p}{q} \cdot \tilde{\mu} \right\rfloor \bmod p$	

Figure 4: BLE construction, main algorithms.

$\text{Upd}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id}, \text{pk})$	$\text{TreeUpdate}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id})$ / subroutine
/ Update the Merkle path of $\text{reg}[t, \text{id}]$ assert $\text{reg}[t, \text{id}][i] = \perp \quad \forall i \in [k]$ / id is not a prefix $\text{reg}[t, \text{id}] := \text{pk}$ $\text{st}' \leftarrow \text{TreeUpdate}^{\text{reg}}(\text{pp}, t, \text{id})$ return st'	/ Maintain invariant: each node is either: / (1) a leaf with value pk or $\text{term}[t]$, or (2) has k children / Assume all $\text{reg}[t, \text{id}]$'s children satisfy invariant / Update siblings and parent of $\text{reg}[t, \text{id}]$ then recurse on parent if $\text{id} = \epsilon$ then / Reached root, stop recursion and return root hash $\text{st}[t] := \text{reg}[t, \epsilon]$ return st $\text{par} := \text{parent}(\text{id})$ if $\text{reg}[t, \text{par}][i] \in \{\perp, \text{term}[t]\} \quad \forall i \in [k]$ then / Case 1: none of the children are assigned $\text{reg}[t, \text{par}][i] := \perp \quad \forall i \in [k]$ / update siblings $\text{reg}[t, \text{par}] := \text{term}[t]$ / update parent else / Case 2: at least one of the children is assigned for $i \in [k]$ s.t. $\text{reg}[t, \text{par}][i] = \perp$ do $\text{reg}[t, \text{par}][i] := \text{term}[t]$ / update siblings $\mathbf{u} := \left(-\tilde{\mathbf{G}}^{-1}(\mathbf{y}_{t, \text{par}}[i]) \right)_{i \in [k]}$ $\text{reg}[t, \text{par}] := \mathbf{y}_{t, \text{par}} := \mathbf{A} \cdot \mathbf{u} \bmod q$ / update parent return $\text{TreeUpdate}^{\text{reg}}(\text{pp}, \text{st}, t, \text{par})$
$\text{WGen}^{\text{reg}}(\text{pp}, \text{st}, t, \text{id})$	
for $j \in [\text{id}]$ do / Denote $\mathbf{y}_{t, \text{str}} := \text{reg}[t, \text{str}]$ for any string str . $\mathbf{u}_{t, \text{id}[j]} := \left(-\tilde{\mathbf{G}}^{-1}(\mathbf{y}_{t, \text{id}[j]}) \right)_{i \in [k]}$ $\text{wit}_t := (\mathbf{u}_{t, \text{id}[j]})_{j \in [\text{id}]}$ return $\text{wit} := (t, \text{wit}_t)$	

Figure 5: BLE construction, data structure algorithms.

Theorem 5.2 (Correctness). *For parameters specified in Table 2, BLE from Figures 4 and 5 is ϵ -correct for some*

$$\epsilon \leq B \cdot \left(2^{-\varphi} + (\hat{\ell} + 2) \cdot 2^{-\varphi n} + \hat{\ell} \cdot 2^{-3\ell_A \varphi n} \right).$$

The proof of Theorem 5.2 can be found in Appendix A.1.

f	free variable in $2^{\mathbb{N}}$ (e.g.: $f = 512$)	φ	$\varphi(f) = f/2$
n	free variable in $\mathbb{N}$ (e.g.: $n = 3$)	b	$\lceil q^{1/(\ell_A+1)} \rceil$
ℓ_A	free variable in $\mathbb{N}$ (e.g.: $\ell_A = 3$)	h	1
B	free variable in $\mathbb{N}$ (e.g.: $B = 30$)	k	3 $\nmid \arg \min_{k \in \mathbb{N}} k / \log k$
σ	free variable > 0	m_A	$n\ell_A$
$\tilde{\sigma}$	free variable > 0	m_B	n
q	free variable given that $q > 2((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b$ $+ 4 \cdot \sqrt{\varphi} \tilde{\sigma} + 2$	p	2
		χ_x, χ_z	$\mathcal{U}(\mathcal{R}_b)$
		χ_r, χ_e	$\mathcal{D}_{\mathcal{R}, \sigma}$
		$\tilde{\chi}_e$	$\mathcal{D}_{\mathcal{R}, \tilde{\sigma}}$

Table 2: Semi-concrete parameter choice for BLE.

Note that the BLE construction BLE from Figures 4 and 5 is non-trivially efficient as the batching overhead is smaller than 1.

$$\begin{aligned}
\lim_{|S| \rightarrow \infty} \frac{|\text{ctxt}_{|S|}|}{|S| \cdot |\text{ctxt}_1|} &= \lim_{|S| \rightarrow \infty} \frac{|(\mathbf{c}_0, \dots, \mathbf{c}_\ell)| + |\mathbf{d}|}{|S| \cdot (|(\mathbf{c}_0, \dots, \mathbf{c}_\ell)| + |\mathbf{d}|)} \\
&= \lim_{|S| \rightarrow \infty} \frac{|(\mathbf{c}_0, \dots, \mathbf{c}_\ell)| + |S| \cdot |\mathbf{d}|}{|S| \cdot (|(\mathbf{c}_0, \dots, \mathbf{c}_\ell)| + |\mathbf{d}|)} \\
&= \frac{|\mathbf{d}|}{|(\mathbf{c}_0, \dots, \mathbf{c}_\ell)| + |\mathbf{d}|} = \frac{|\mathbf{d}|}{|\text{ctxt}_1|} < 1
\end{aligned}$$

and clearly $|\text{ctxt}_{|S|}| \leq |\text{ctxt}_{|S'|}|$ whenever $S \subseteq S'$.

As an efficiency optimisation, one can omit the least significant bits of $\mathcal{R}_q$ in the ciphertexts, because these are noisy/random anyway. This optimisation has been noted in [34] and is widely used, see Kyber [11, Footnote 2]. We formalise this in the following corollary. We note here, that bit-dropping can only increase the security of our encryption, since an adversary can simply ignore the least significant bits, if that were advantageous for an attack.

Corollary 5.3 (Bit-dropping). *Let $D \in \mathbb{N}_0$ be a parameter and for the parameters specified in Table 2 assume that additionally*

$$q > 2 \cdot ((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b + 4 \cdot \sqrt{\varphi} \tilde{\sigma} + 2 + 2^{D+1} \cdot (1 + (\hat{\ell} + 1) \cdot \varphi \cdot b/2)$$

holds. Following Figures 4 and 5, consider a modified algorithm for encryption Enc' that runs $(\mathbf{c}_0, \dots, \mathbf{c}_\ell, \mathbf{d}) \leftarrow \text{Enc}(\text{pp}, \text{st}, S, \text{id}, \mathbf{m})$ and outputs $(\mathbf{c}'_0, \dots, \mathbf{c}'_\ell, \mathbf{d}')$, where $\mathbf{c}'_i = 2^D \cdot \lfloor \mathbf{c}_i / 2^D \rfloor$ and $\mathbf{d}' = 2^D \cdot \lfloor \mathbf{d} / 2^D \rfloor$. In other words, in BLE' , the encryption Enc' zeroes out the D least significant bits of all ciphertext components.

Then BLE' from Figures 4 and 5 with modified Enc' is ϵ' -correct for some

$$\epsilon' \leq B \cdot \left(2^{-\varphi} + (\hat{\ell} + 2) \cdot 2^{-\varphi n} + \hat{\ell} \cdot 2^{-3\ell_A \varphi n} \right).$$

The proof of Corollary 5.3 is available in Appendix A.2.

As a final remark on the correctness of bit-dropping, observe that we could treat $\mathbf{c}$ and $\mathbf{d}$ separately, using different choices of D , as these have different amounts of noise. However, since the required batch size $B \approx 30$ for RBE is only a small fraction of $k \cdot m_A$ (typically ≥ 300 in our setting), we avoided the additional complexity. For applications with large B , the differentiation of $\mathbf{c}$ and $\mathbf{d}$ may be worthwhile.

Remark 5.4 (Heuristic correctness bound). Dropping small constants from the lower bound of q (Equation (A.3) in Theorem 5.2), and assuming the two summands are about the same size, we obtain the approximate requirement of

$$\begin{aligned}
\tilde{\sigma} &\approx 1.5 \cdot \hat{\ell} \cdot \varphi \cdot \ell_A \cdot n \cdot \sigma \cdot b, \\
q &\approx b^{\ell_A+1} \gtrsim 6 \cdot \hat{\ell} \cdot \varphi^{3/2} \cdot \ell_A \cdot n \cdot \sigma \cdot b.
\end{aligned}$$

Even this lower bound of q is likely to be an overestimate due to how the norm of a product of low-norm vectors is bounded. A standard heuristic is to remove a factor of $\sqrt{(3\ell_A + 1)n \cdot \varphi}$ (i.e. square-root of the ciphertext $\mathbb{Z}_q$ -dimension). This results in an approximate requirement of

$$\begin{aligned}
\tilde{\sigma} &\approx 0.9 \cdot \hat{\ell} \cdot \varphi^{1/2} \cdot \sqrt{\ell_A \cdot n \cdot \sigma \cdot b}, \\
q &\approx b^{\ell_A+1} \gtrsim 3.5 \cdot \hat{\ell} \cdot \varphi \cdot \sqrt{\ell_A \cdot n \cdot \sigma \cdot b}.
\end{aligned}$$

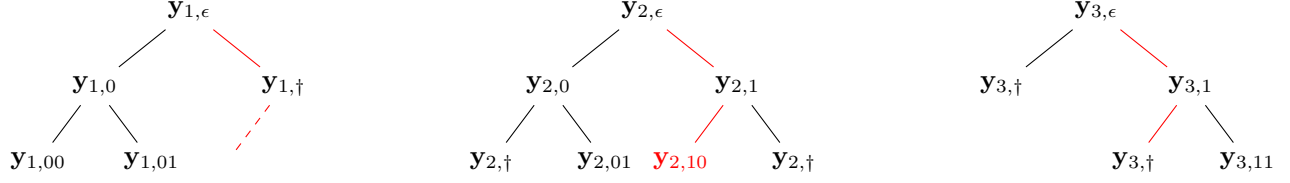


Figure 6: Some possible configurations of reg in relation to a given identity $\text{id} = 10$ (assuming $(k = 2)$ -ary identities). In the first tree, only $\text{id} = 00$ and 01 are registered, thus the path 10 ends prematurely at node 1 with a terminator label. In the second tree, $\text{id} = 10$ is registered. In the third tree, only $\text{id} = 11$ is registered, thus the path 10 ends with a terminator label.

Next, we state and prove the security of our BLE scheme from Figures 4 and 5. Similar to [8], we state the security theorem under the Leaky LWE assumption. In Corollary 5.6, we plug in the LWE-to-LLWE reduction from Lemma 3.7 to state parameter requirements solely under the LWE assumption.

Theorem 5.5 (IND\$-CPA). *Let $km_A \geq m_B$. Let*

- $\text{lwepar}(\lambda) = (\mathcal{R}, m_B, n, q, \chi_x, \chi_z)$ and
- $\text{llwepar}(\lambda) = (\mathcal{R}, n, km_A + B, q, B, \chi_r, \chi_e, \tilde{\chi}_e, \mathcal{L})$

be LWE and Leaky LWE parameters, respectively, where $\mathcal{L}$ consists of matrices $\mathbf{L}$ with $|\mathbf{L}| \leq (km_A + n)B$ and $\|\mathbf{L}\|_\infty \leq b^h/2$. Suppose that $\chi_e + \bar{\chi}_e$ is statistically δ -close to $\tilde{\chi}_e$ and $\delta = \text{negl}(\lambda)$. For any PPT adversary $\mathcal{A}$ against the IND\$-CPA security of BLE in Figures 4 and 5 making at most Q_{KGen} queries to KGenO , there exist PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ with running time roughly that of $\text{Exp-IND\$-CPA}_{\text{BLE}, \mathcal{A}}^b(1^\lambda)$, such that $\text{Adv-IND\$-CPA}_{\text{BLE}, \mathcal{A}}(\lambda)$ is at most

$$\begin{aligned} & Q_{\text{KGen}} \cdot \text{Adv-LWE}_{\text{lwepar}, \mathcal{A}_1}(\lambda) \\ & + (\hat{\ell} + 1) \cdot \text{Adv-LLWE}_{\text{llwepar}, \mathcal{A}_2}(\lambda) \\ & + B \cdot \delta \end{aligned}$$

The proof of Theorem 5.5 follows the same strategy as in [8], with two major differences: (1) Since the adversary is now allowed to specify a set S of sub-registries for encryption, the given identity id may be registered in some sub-registries but not others. In particular, for those sub-registries in which id is not registered, when traversing the trees of different sub-registries along the path indexed by id , we may encounter terminator labels at different depths. An example is described in Figure 6. (2) Due to the use of approximate gadgets, our proof relies on the Leaky LWE assumption [31], which captures both *secret*- and error-leakage, instead of the error-leakage LWE assumption [8]. The proof of Theorem 5.5 can be found in Appendix A.3.

Applying the reduction from LWE to Leaky LWE (Lemma 3.7), we have the following corollary.

Corollary 5.6 (IND\$-CPA). *Consider the parameters specified in Table 2. Let $\sigma^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathbb{Z}^{\varphi(kn\ell_A + B)})$, $\sigma, \bar{\sigma} \geq \sqrt{2} \cdot \eta_\epsilon(\mathbb{Z}^{\varphi B})$ for some $\epsilon \leq \text{negl}(\lambda)$ be such that*

$$\frac{(k\ell_A + 1)nB\varphi^2b^2}{4\bar{\sigma}^2} + \frac{1}{\sigma^2} \leq \frac{1}{2\eta_\epsilon(\mathbb{Z}^{\varphi((1+k\ell_A)n+B)})^2 + (\sigma^*)^2}.$$

Let

- $\text{lwepar}_1(\lambda) = (\mathcal{R}, n, n, q, \mathcal{U}(\mathcal{R}_b), \mathcal{U}(\mathcal{R}_b))$ and
- $\text{lwepar}_2(\lambda) = (\mathcal{R}, n, kn\ell_A + B, q, \mathcal{D}_{\mathcal{R}, \sigma^*}, \mathcal{D}_{\mathcal{R}, \sigma^*})$

be LWE parameters. For any PPT adversary $\mathcal{A}$ against the IND\$-CPA security of BLE in Figures 4 and 5 making at most Q_{KGen} queries to KGenO , there exist PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ with running time roughly that of $\text{Exp-IND\$-CPA}_{\text{BLE}, \mathcal{A}}^b(1^\lambda)$, such that $\text{Adv-IND\$-CPA}_{\text{BLE}, \mathcal{A}}(\lambda)$ is at most

$$\begin{aligned} & Q_{\text{KGen}} \cdot \text{Adv-LWE}_{\text{lwepar}_1, \mathcal{A}_1}(\lambda) \\ & + (\hat{\ell} + 1) \cdot \text{Adv-LWE}_{\text{lwepar}_2, \mathcal{A}_2}(\lambda) \\ & + 4Q_{\text{KGen}} \cdot \epsilon / (1 - \epsilon) + 16B\epsilon. \end{aligned}$$

Proof. Observe that the choice of lwepar_1 and lwepar_2 and the imposed relations on $\sigma, \bar{\sigma}, \sigma^*$ are exactly as required by Lemma 3.7. Hence, it suffices to justify $\delta \leq 16\epsilon$ for δ as in Theorem 5.5. This follows by Corollary 3.4, which asserts that the sum $\chi_e + \bar{\chi}_e$ is statistically 16ϵ -close to $\tilde{\chi}_e$, since χ_e and $\bar{\chi}_e$ are discrete Gaussians of parameter $\sigma, \bar{\sigma} \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^\varphi)$ by assumption. $\square$

6. Implementation and Benchmarks

In this section, we describe our strategy to find suitable parameters, provide insights on our prototype, and elaborate on its benchmarks.

6.1. Parameter Generation

First, notice that we fix some parameters in Table 2. As pointed out in [9], an RBE scheme should support 2^{256} identities for collision-resistance. Thus, we pick $\hat{\ell}$ so that $3^{\hat{\ell}} > 2^{256}$ resulting in $\hat{\ell} = 162$. To support 2^{30} users, we require the batch size B to be 30. We recall Theorem 5.2 to guarantee correctness of our construction, requiring $q > 2((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b + 4 \cdot \sqrt{\varphi\tilde{\sigma}} + 2$ (ignoring bit-dropping). For security, we require LWE to be hard for the following sets of parameters.

- $\text{lwepar}_1(\lambda) = (\mathcal{R}, n, n, q, \mathcal{U}(\mathcal{R}_b), \mathcal{U}(\mathcal{R}_b))$
- $\text{lwepar}_2(\lambda) = (\mathcal{R}, n, kn\ell_A + B, q, \mathcal{D}_{\mathcal{R}, \sigma^*}, \mathcal{D}_{\mathcal{R}, \sigma^*})$

Furthermore, Corollary 5.6 requires the following bound to hold for some $\sigma^* \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^{\varphi(kn\ell_A + B)})$:

$$\frac{(k\ell_A + 1)nB\varphi^2b^2}{4\tilde{\sigma}^2} + \frac{1}{\sigma^2} \leq \frac{1}{2\eta_\epsilon(\mathbb{Z}^{\varphi((1+k\ell_A)n+B)})^2 + (\sigma^*)^2}.$$

Given these constraints, we optimise for a minimal ciphertext size considering practicality of our parameters. We pick ℓ_A as small as possible to minimise the ciphertext size, while ensuring that $q < 2^{64}$ for efficient arithmetic. Concretely, this means picking $\ell_A = 2$, i.e. $b \approx q^{1/3}$.

The ring degree φ and module rank n are picked together to ensure a large enough (plain) LWE secret dimension. Following Kyber, we pick $\varphi = 256$ and choose n to be the smallest n such that LWE is hard enough.

Given the above, we extract two variables to optimise.

$$\frac{((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi \cdot n \cdot \sigma \cdot b}{2\tilde{\sigma}} \rightarrow 1, \quad (6.1)$$

$$\frac{\sigma \cdot \sqrt{(k \cdot \ell_A + 1) \cdot n \cdot B \cdot \varphi \cdot b}}{\sqrt{2}\tilde{\sigma}} \rightarrow 1. \quad (6.2)$$

Equation (6.1) is close to 1 if both additive terms lower bounding q are almost of the same size, which allows for small q and therefore a compact ciphertext. Equation (6.2) is close to 1 if the additive terms, which lower bound σ^* , are of a similar size. A large σ^* increases the hardness of LWE for lwepar_2 , which is generally the weaker LWE instance.

Considering all constraints of our proofs, we use the lattice-estimator⁵ [35] in a Sage script to estimate the hardness of LWE and optimise the ciphertext size using both indicators described above. In our case, a ciphertext against a set of B sub-registries to an identity of length ℓ consists of $(3\ell_A\hat{\ell} + 1)n + B$ elements of $\mathcal{R}_q$. Each $\mathcal{R}_q$ element requires $\varphi \log q$ bits to represent. We aim for 128-bit security and find the following parameters using Corollary 5.3 to enable bit-dropping for the ciphertext to reduce its size further.

$$\begin{aligned} \varphi &= 256, & q &= 281,474,976,694,273 \approx 2^{47.99} \\ n &= 7, & \tilde{\sigma} &= 2^{35}, & D &= 16, & b &= 65,536, & \sigma &= 16.0. \end{aligned}$$

These parameters result in a ciphertext size of 7.0 MB and enable the application of optimised 64-bit arithmetic.

6.2. Implementation

To confirm the scalability and performance of our RBE construction, we implement it. Our open-source prototype benefits from the memory-safety guarantees and efficiency characteristics inherent with a Rust implementation. The prototype can be found in the following repository:

https://github.com/jnsiemer/scalable_rbe_prototype

Our implementation provides modularity and an extensive documentation to encourage experimentation and extension. At this stage of development, we did not emphasise side-channel security and explicitly discourage any production-grade deployment.

⁵Concretely, we use the commit '352ddaf'.

	φ	Time (milliseconds)						Size (MegaBytes)		
		Setup	KeyGen	Upd	Enc	WGen	Dec	ctxt	st	reg
[8]	256	150 · 49	7.72	5 · 1,565	150 · 2.95	5 · 0.061	5 · 6.27	150 · 49	8	150 · 367
Our RBE	256	9.98	0.12	33.24	15.56	3.38	10.43	7.0	0.34	1,626.5

Table 3: Benchmarks for implementations of RBE with $N = 2^{30}$ and $|\mathcal{ID}| = \{0, 1\}^{256}$. The size of pp is 64 Bytes. The size of the registry is computed for $\tilde{n} = 1000$ registered users and the scalars represent the LE to RBE compiler proposed in [8]. Multi-threaded benchmarks are reported in bold.

Compared to our algorithm descriptions in Figure 4 and 5, we utilise the following more significant optimisations in our implementation.

- Rather than computing $-\tilde{\mathbf{G}}^{-1}(\mathbf{y}_{t,\text{par}}||i)$ for all $k \cdot 162$ nodes on the Merkle-path from ϵ to id, we store this value directly for all levels except the root. This removes the need for repeated computation in Upd and WGen. Additionally, we optimise storage cost by storing a pointer to $\mathbf{y}_{t,\text{term}}$ rather than storing $\mathbf{y}_{t,\text{term}}$ hundreds of times per tree and combine writes to the database into a single request.
- We utilize the Number-Theoretic Transform (NTT) supporting hardware acceleration for efficient matrix multiplication using the library *concrete-ntt*.⁶
- Following best practices from Kyber and ML-KEM, we compress the ciphertext by dropping lower-order bits of each coefficient. Furthermore, we use two random 256 bit seeds as pp and generate the matrices $\mathbf{A} \in \mathcal{R}_q^{n \times m_A}$ and $\mathbf{B} \in \mathcal{R}_q^{n \times m_B}$ from these seeds.
- As several algorithms in Figure 4 and 5 can be parallelised, we utilise multi-threading in the implementation of the encryption algorithm.

Sampling from a discrete Gaussian distribution is a major concern in terms of efficiency and constant-time security in lattice-based cryptography. Our RBE construction requires more than $k \cdot m_A \cdot \varphi \cdot (\hat{\ell} + 2) \gtrsim 1.7 \cdot 10^6$ discrete Gaussian samples of varying widths per execution of Enc. To generate these values efficiently, we implement inverse transform sampling and Algorithm 2 from [36] to sample discrete Gaussian distributed values. The latter algorithm only allows sampling of discrete Gaussian values where the Gaussian parameter is in an interval of size $[s, \sqrt{5}s]$. As Algorithm 2 of [36] significantly outperforms several other discrete Gaussian samplers (on a CPU) for large Gaussian parameters, we consider this lack of precise control over the exact parameter to be an acceptable trade-off.

A shortcoming of our prototype is the simplified implementation of the update algorithm Upd. Currently, our code resembles the algorithm described in Figure 5 closely, but it does not implement the merging operation of trees as described in the generic transform from LE to RBE [8]. This transform requires insertion of public keys into an empty tree. Once two trees contain the same number of public keys, these trees are merged to bound the number of witness updates to $\log N$ per identity. As we register users in one bulk, we insert their public keys directly into the tree, in which they would end up after registering the simulated amount of identities in our benchmarks.

To estimate the cost of merging two Merkle trees, we rely on the assumption that our k -ary identity string $\text{id} \leftarrow H(\text{"real_identity_string"})$ results from hashing the real identity string. Thus, it is reasonable to operate under the assumption that our identity strings id are distributed close to uniform in $[k]^\ell$. The cost of merging two k -ary trees mostly depends on the shared length of prefixes of their identity strings, i.e. the number of nodes in the k -ary trie spanned by the identities. Every node of the spanned trie needs to be updated by recomputing the SIS-based hash function utilised in the Merkle tree. This operation will dominate the runtime as subtrees can be moved by reassigning pointers. Assuming that identities are distributed uniformly, the expected depth of a k -ary trie containing $\tilde{n}$ identities is $\mathcal{O}(\log_k(\tilde{n}))$. A trie of depth $\mathcal{O}(\log_k(\tilde{n}))$ contains at most $\mathcal{O}(\tilde{n})$ nodes and merging two trees with $\mathcal{O}(\tilde{n})$ nodes in a depth-first approach requires $\mathcal{O}(\tilde{n})$ operations. Thus, the expected cost of merging two trees is dominated by the runtime of $\mathcal{O}(\tilde{n})$ computations of the SIS-based hash function, which amortises over the logarithmic number of times that this operation has to be performed per registered user.

Finally, we point out a few trade-offs, which developers can optimise for their specific applications and conditions.

- In Upd, we combine all writes to a single database query. This approach could be extended to reads of database values.
- The matrix $\tilde{\mathbf{G}}$ is sparse. To optimise the encryption algorithm further, one could employ a specific matrix multiplication algorithm for sparse matrices.
- In addition to Enc, the execution time of WGen and Dec could be sped up using multi-threading.
- For static parameter choices, we expect that further fine-tuning of memory layout can decrease runtimes.

None of the mentioned trade-off seems to dominate the execution time of our algorithms. Thus, we decided to keep the implementation as general as possible.

⁶<https://crates.io/crates/concrete-ntt>

6.3. Benchmarks

We use a popular Rust library called *criterion*⁷ for benchmarking and employ a key-value database called *sled*⁸ to manage the registry. All benchmarks were conducted on a personal laptop with an Intel Core Ultra 7 165U 4.9 GHz CPU with 32 GB of RAM running Ubuntu 24.04.3 with kernel 6.8.0-87-generic never utilising more than 14 threads simultaneously. The mean of each measurement for this setup is reported in Table 3. The benchmarks for our construction only exclude the computation of strictly precomputable values such as tables for discrete Gaussian samplers, i.e. values which would be hard-coded in a less-experimental framework.

As our work is based on the same framework as [8], our prototype is necessarily compared to their implementation of LE. We present measurements for our prototype in Table 3 alongside theirs, considering their most aggressive choice of parameters. Unfortunately, there is no information available regarding the security level of their parameters. Therefore, we only compare against their best parameters. They set $\ell = 50$ for identity-arity $k = 2$, which would require them to lengthen their identity strings by a factor of 5 to be comparable to our setting (as $3^{162} \gtrsim 2^{256}$). Furthermore, their implementation uses one Merkle tree, which implies a factor of 30 to accomodate the LE-to-RBE transform to comply with our RBE setting with $N = 2^{30}$ users. We represent these adjustments by a multiplication factor for each reported runtime and size in Table 3. The correctness of our scaling can be verified using [9], which reports a ciphertext size of 7.2 GB for their scheme in our setting indicating a factor of 150.

Other than using different choices of databases, there are a few further differences between measurement methods and implementation. Our prototype only uses multi-threading to speed up the encryption algorithm Enc. Döttling et al. [8] additionally parallelise Upd, Dec, and WGen. Furthermore, they exclude certain actions such as the setup of their database and the generation of any randomness for encryption from their measurements. Using precomputed randomness, especially for more than $1.7 \cdot 10^6$ discrete Gaussian samples, seems unlikely in practice as temporary storage of ephemeral secrets not only poses a security risk but also requires a significant amount of time to retrieve from memory. Thus, we include the generation of random values in our reported benchmarks for all algorithms.

Acknowledgements. We thank the anonymous reviewers for insightful comments that improved the presentation of this paper. The research of Michael Klooß is supported by KASTEL Security Research Labs. The research of Russell W. F. Lai and Monisha Swarnakar is supported by the Research Council of Finland grant 358951. The research of Jan Niklas Siemer was partially funded by the EPSRC Research Council, part of the EPSRC/UKRI Horizon Europe Grant Studentship, Grant Ref: EP/Y02432X/1.

References

- [1] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *CRYPTO’84*, G. R. Blakley and D. Chaum, Eds., ser. LNCS, vol. 196, Springer, Berlin, Heidelberg, Aug. 1984, pp. 47–53. DOI: [10.1007/3-540-39568-7_5](https://doi.org/10.1007/3-540-39568-7_5). 1
- [2] D. Boneh and M. K. Franklin, “Identity based encryption from the Weil pairing,” *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003. 1
- [3] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” in *40th ACM STOC*, R. E. Ladner and C. Dwork, Eds., ACM Press, May 2008, pp. 197–206. DOI: [10.1145/1374376.1374407](https://doi.org/10.1145/1374376.1374407). 1, 4, 9
- [4] S. Garg, M. Hajiabadi, M. Mahmoody, and A. Rahimi, “Registration-based encryption: Removing private-key generator from IBE,” in *TCC 2018, Part I*, A. Beimel and S. Dziembowski, Eds., ser. LNCS, vol. 11239, Springer, Cham, Nov. 2018, pp. 689–718. DOI: [10.1007/978-3-030-03807-6_25](https://doi.org/10.1007/978-3-030-03807-6_25). 1, 2
- [5] S. Garg, M. Hajiabadi, M. Mahmoody, A. Rahimi, and S. Sekar, “Registration-based encryption from standard assumptions,” in *PKC 2019, Part II*, D. Lin and K. Sako, Eds., ser. LNCS, vol. 11443, Springer, Cham, Apr. 2019, pp. 63–93. DOI: [10.1007/978-3-030-17259-6_3](https://doi.org/10.1007/978-3-030-17259-6_3). 2
- [6] K. Cong, K. Eldefrawy, and N. P. Smart, “Optimizing registration based encryption,” in *Cryptography and Coding*, M. B. Paterson, Ed., Cham: Springer International Publishing, 2021, pp. 129–157, ISBN: 978-3-030-92641-0. 2
- [7] N. Glaeser, D. Kolonelos, G. Malavolta, and A. Rahimi, “Efficient registration-based encryption,” in *ACM CCS 2023*, W. Meng, C. D. Jensen, C. Cremers, and E. Kirda, Eds., ACM Press, Nov. 2023, pp. 1065–1079. DOI: [10.1145/3576915.3616596](https://doi.org/10.1145/3576915.3616596). 2
- [8] N. Döttling, D. Kolonelos, R. W. F. Lai, C. Lin, G. Malavolta, and A. Rahimi, “Efficient laconic cryptography from learning with errors,” in *EUROCRYPT 2023, Part III*, C. Hazay and M. Stam, Eds., ser. LNCS, vol. 14006, Springer, Cham, Apr. 2023, pp. 417–446. DOI: [10.1007/978-3-031-30620-4_14](https://doi.org/10.1007/978-3-031-30620-4_14). 2–5, 7–10, 13, 15, 16
- [9] D. Fiore, D. Kolonelos, and P. de Perthuis, “Cuckoo commitments: Registration-based encryption and key-value map commitments for large spaces,” in *ASIACRYPT 2023, Part V*, J. Guo and R. Steinfeld, Eds., ser. LNCS, vol. 14442, Springer, Singapore, Dec. 2023, pp. 166–200. DOI: [10.1007/978-981-99-8733-7_6](https://doi.org/10.1007/978-981-99-8733-7_6). 2, 3, 14, 16
- [10] S. Hohenberger, G. Lu, B. Waters, and D. J. Wu, “Registered attribute-based encryption,” in *EUROCRYPT 2023, Part III*, C. Hazay and M. Stam, Eds., ser. LNCS, vol. 14006, Springer, Cham, Apr. 2023, pp. 511–542. DOI: [10.1007/978-3-031-30620-4_17](https://doi.org/10.1007/978-3-031-30620-4_17). 2, 3

⁷<https://crates.io/crates/criterion>

⁸<https://crates.io/crates/sled>

- [11] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, “CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM,” in *2018 IEEE European Symposium on Security and Privacy*, IEEE Computer Society Press, Apr. 2018, pp. 353–367. DOI: [10.1109/EuroSP.2018.00032](https://doi.org/10.1109/EuroSP.2018.00032). 3, 12
- [12] B. Libert, S. Ling, K. Nguyen, and H. Wang, “Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors,” in *EUROCRYPT 2016, Part II*, M. Fischlin and J.-S. Coron, Eds., ser. LNCS, vol. 9666, Springer, Berlin, Heidelberg, May 2016, pp. 1–31. DOI: [10.1007/978-3-662-49896-5_1](https://doi.org/10.1007/978-3-662-49896-5_1). 3, 4
- [13] D. Catalano and D. Fiore, “Vector commitments and their applications,” in *PKC 2013*, K. Kurosawa and G. Hanaoka, Eds., ser. LNCS, vol. 7778, Springer, Berlin, Heidelberg, 2013, pp. 55–72. DOI: [10.1007/978-3-642-36362-7_5](https://doi.org/10.1007/978-3-642-36362-7_5). 3
- [14] J. Dujmovic, G. Malavolta, and W. Qi, “Registration-based encryption in the plain model,” in *PKC 2025, Part I*, T. Jager and J. Pan, Eds., ser. LNCS, vol. 15674, Springer, Cham, May 2025, pp. 194–226. DOI: [10.1007/978-3-031-91820-9_7](https://doi.org/10.1007/978-3-031-91820-9_7). 3
- [15] C. Freitag, B. Waters, and D. J. Wu, “How to use (plain) witness encryption: Registered ABE, flexible broadcast, and more,” in *CRYPTO 2023, Part IV*, H. Handschuh and A. Lysyanskaya, Eds., ser. LNCS, vol. 14084, Springer, Cham, Aug. 2023, pp. 498–531. DOI: [10.1007/978-3-031-38551-3_16](https://doi.org/10.1007/978-3-031-38551-3_16). 3
- [16] N. Attrapadung and J. Tomida, “A modular approach to registered ABE for unbounded predicates,” in *CRYPTO 2024, Part III*, L. Reyzin and D. Stebila, Eds., ser. LNCS, vol. 14922, Springer, Cham, Aug. 2024, pp. 280–316. DOI: [10.1007/978-3-031-68382-4_9](https://doi.org/10.1007/978-3-031-68382-4_9). 3
- [17] P. Datta, T. Pal, and S. Yamada, “Registered FE beyond predicates: (attribute-based) linear functions and more,” in *ASIACRYPT 2024, Part I*, K.-M. Chung and Y. Sasaki, Eds., ser. LNCS, vol. 15484, Springer, Singapore, Dec. 2024, pp. 65–104. DOI: [10.1007/978-981-96-0875-1_3](https://doi.org/10.1007/978-981-96-0875-1_3). 3
- [18] P. Branco, R. W. F. Lai, M. Maitra, G. Malavolta, A. Rahimi, and I. K. Y. Woo, “Traitor tracing without trusted authority from registered functional encryption,” in *ASIACRYPT 2024, Part III*, K.-M. Chung and Y. Sasaki, Eds., ser. LNCS, vol. 15486, Springer, Singapore, Dec. 2024, pp. 33–66. DOI: [10.1007/978-981-96-0891-1_2](https://doi.org/10.1007/978-981-96-0891-1_2). 3
- [19] G. Lu, B. Waters, and D. J. Wu, “Multi-authority registered attribute-based encryption,” in *EUROCRYPT 2025, Part III*, S. Fehr and P.-A. Fouque, Eds., ser. LNCS, vol. 15603, Springer, Cham, May 2025, pp. 3–33. DOI: [10.1007/978-3-031-91131-6_1](https://doi.org/10.1007/978-3-031-91131-6_1). 3
- [20] R. Garg, G. Lu, B. Waters, and D. J. Wu, “Reducing the CRS size in registered ABE systems,” in *CRYPTO 2024, Part III*, L. Reyzin and D. Stebila, Eds., ser. LNCS, vol. 14922, Springer, Cham, Aug. 2024, pp. 143–177. DOI: [10.1007/978-3-031-68382-4_5](https://doi.org/10.1007/978-3-031-68382-4_5). 3
- [21] S. Garg, M. Hajiabadi, D. Kolonelos, A. Kothapalli, and G.-V. Policharla, “A framework for witness encryption from linearly verifiable SNARKs and applications,” in *CRYPTO 2025, Part III*, Y. T. Kalai and S. F. Kamara, Eds., ser. LNCS, vol. 16002, Springer, Cham, Aug. 2025, pp. 504–539. DOI: [10.1007/978-3-032-01881-6_16](https://doi.org/10.1007/978-3-032-01881-6_16). 3
- [22] H. Wee and D. J. Wu, “Unbounded distributed broadcast encryption and registered ABE from succinct LWE,” in *CRYPTO 2025, Part III*, Y. T. Kalai and S. F. Kamara, Eds., ser. LNCS, vol. 16002, Springer, Cham, Aug. 2025, pp. 204–235. DOI: [10.1007/978-3-032-01881-6_7](https://doi.org/10.1007/978-3-032-01881-6_7). 3
- [23] J. Champion, Y.-C. Hsieh, and D. J. Wu, “Registered ABE and adaptively-secure broadcast encryption from succinct LWE,” in *CRYPTO 2025, Part III*, Y. T. Kalai and S. F. Kamara, Eds., ser. LNCS, vol. 16002, Springer, Cham, Aug. 2025, pp. 3–34. DOI: [10.1007/978-3-032-01881-6_1](https://doi.org/10.1007/978-3-032-01881-6_1). 3
- [24] M. Ajtai, “Generating hard instances of lattice problems (extended abstract),” in *28th ACM STOC*, ACM Press, May 1996, pp. 99–108. DOI: [10.1145/237814.237838](https://doi.org/10.1145/237814.237838). 4
- [25] M. R. Albrecht and R. W. F. Lai, “Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices,” in *CRYPTO 2021, Part II*, T. Malkin and C. Peikert, Eds., ser. LNCS, vol. 12826, Virtual Event: Springer, Cham, Aug. 2021, pp. 519–548. DOI: [10.1007/978-3-030-84245-1_18](https://doi.org/10.1007/978-3-030-84245-1_18). 6
- [26] D. Micciancio and O. Regev, “Worst-case to average-case reductions based on Gaussian measures,” in *45th FOCS*, IEEE Computer Society Press, Oct. 2004, pp. 372–381. DOI: [10.1109/FOCS.2004.72](https://doi.org/10.1109/FOCS.2004.72). 6
- [27] D. Micciancio and C. Peikert, “Trapdoors for lattices: Simpler, tighter, faster, smaller,” in *EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds., ser. LNCS, vol. 7237, Springer, Berlin, Heidelberg, Apr. 2012, pp. 700–718. DOI: [10.1007/978-3-642-29011-4_41](https://doi.org/10.1007/978-3-642-29011-4_41). 6, 7
- [28] N. Genise, D. Micciancio, C. Peikert, and M. Walter, “Improved discrete gaussian and subgaussian analysis for lattice cryptography,” in *PKC 2020, Part I*, A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, Eds., ser. LNCS, vol. 12110, Springer, Cham, May 2020, pp. 623–651. DOI: [10.1007/978-3-030-45374-9_21](https://doi.org/10.1007/978-3-030-45374-9_21). 6
- [29] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *37th ACM STOC*, H. N. Gabow and R. Fagin, Eds., ACM Press, May 2005, pp. 84–93. DOI: [10.1145/1060590.1060603](https://doi.org/10.1145/1060590.1060603). 6
- [30] A. Langlois and D. Stehlé, “Worst-case to average-case reductions for module lattices,” *DCC*, vol. 75, no. 3, pp. 565–599, 2015. DOI: [10.1007/s10623-014-9938-4](https://doi.org/10.1007/s10623-014-9938-4). 6
- [31] R. W. F. Lai, M. Swarnakar, and I. K. Y. Woo, “Leaky LWE: Learning with errors with semi-adaptive secret- and error-leakage,” *CiC*, vol. 2, no. 3, p. 21, 2025. DOI: [10.62056/ah89ksuc2](https://doi.org/10.62056/ah89ksuc2). 6, 7, 10, 13
- [32] Y. Chen, N. Genise, and P. Mukherjee, “Approximate trapdoors for lattices and smaller hash-and-sign signatures,” in *ASIACRYPT 2019, Part III*, S. D. Galbraith and S. Moriai, Eds., ser. LNCS, vol. 11923, Springer, Cham, Dec. 2019, pp. 3–32. DOI: [10.1007/978-3-030-34618-8_1](https://doi.org/10.1007/978-3-030-34618-8_1). 7
- [33] E. Fujisaki and T. Okamoto, “Secure integration of asymmetric and symmetric encryption schemes,” *Journal of Cryptology*, vol. 26, no. 1, pp. 80–101, Jan. 2013. DOI: [10.1007/s00145-011-9114-1](https://doi.org/10.1007/s00145-011-9114-1). 9
- [34] C. Peikert, “Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract,” in *41st ACM STOC*, M. Mitzenmacher, Ed., ACM Press, 2009, pp. 333–342. DOI: [10.1145/1536414.1536461](https://doi.org/10.1145/1536414.1536461). 12
- [35] M. R. Albrecht, R. Player, and S. Scott, *On the concrete hardness of learning with errors*, Cryptology ePrint Archive, Report 2015/046, 2015. 14

- [36] D. Micciancio and M. Walter, “Gaussian sampling over the integers: Efficient, generic, constant-time,” in *CRYPTO 2017, Part II*, J. Katz and H. Shacham, Eds., ser. LNCS, vol. 10402, Springer, Cham, Aug. 2017, pp. 455–485. DOI: [10.1007/978-3-319-63715-0_16](https://doi.org/10.1007/978-3-319-63715-0_16). 15
- [37] Y. T. Kalai and S. F. Kamara, Eds., *CRYPTO 2025, Part III*, vol. 16002, LNCS, Springer, Cham, Aug. 2025
- [38] L. Reyzin and D. Stebila, Eds., *CRYPTO 2024, Part III*, vol. 14922, LNCS, Springer, Cham, Aug. 2024
- [39] C. Hazay and M. Stam, Eds., *EUROCRYPT 2023, Part III*, vol. 14006, LNCS, Springer, Cham, Apr. 2023

Appendix A. Omitted proofs

A.1. Proof of Theorem 5.2

Proof. Recall that an honestly generated ciphertext takes the form:

$$\begin{aligned} \mathbf{c}_j^T &= (\mathbf{r}_j, \mathbf{r}_{j+1})^T \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} + \mathbf{e}_j^T \bmod q \in \mathcal{R}_q^{3\ell_A n}, \\ \mathbf{c}_\ell^T &= \mathbf{r}_\ell^T \cdot \mathbf{B} + \mathbf{e}_\ell^T \bmod q \in \mathcal{R}_q^n, \\ \mathbf{d}^T &= \mathbf{r}_0^T \cdot \mathbf{Y}_{S,\epsilon} + \tilde{\mathbf{e}}^T + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m}^T \bmod q \in \mathcal{R}_q^S \end{aligned}$$

where we already plugged in $k = 3$.

Consider decrypting with a membership witness $\text{wit} = (t, (\mathbf{u}_{t,\text{id}[j]})_{j \in [\ell]})$ and secret key $\mathbf{x}_{t,\text{id}}$ satisfying $\mathbf{y}_{t,\text{id}} = \mathbf{y}_{t,\text{id}[:\ell]} = \mathbf{B}\mathbf{x}_{t,\text{id}} + \mathbf{z}_{t,\text{id}} \bmod q$. By the choice of the distributions $\chi_x, \chi_z, \chi_r, \chi_e, \tilde{\chi}_e$ and by Lemma 3.3, we have

$$\begin{aligned} \|\mathbf{x}_{t,\text{id}}\|_\infty, \|\mathbf{z}_{t,\text{id}}\|_\infty &\leq b/2, \\ \Pr[\|\mathbf{r}_j\|_2 > \sqrt{\varphi n} \cdot \sigma] &\leq 2^{-\varphi n}, \quad \forall j \in \{0, 1, \dots, \ell\}, \\ \Pr[\|\mathbf{e}_j\|_2 > \sqrt{3\ell_A \varphi n} \cdot \sigma] &\leq 2^{-3\ell_A \varphi n}, \quad \forall j \in [\ell] \\ \Pr[\|\mathbf{e}_\ell\|_2 > \sqrt{\varphi n} \cdot \sigma] &\leq 2^{-\varphi n}, \\ \Pr[\|\tilde{\mathbf{e}}_t\|_2 > \sqrt{\varphi} \tilde{\sigma}] &\leq 2^{-\varphi} \quad \forall t \in S. \end{aligned}$$

Recall that $\mathbf{u}_{t,\text{id}[j]} = \left(-\tilde{\mathbf{G}}^{-1}(\mathbf{y}_{t,\text{id}[j]}\|i)\right)_{i \in [k]}$ satisfies $\|\mathbf{u}_{t,\text{id}[j]}\|_\infty \leq b/2$. Let

$$\mathbf{f}_{t,j+1} = \mathbf{y}_{t,\text{id}[j+1]} - \tilde{\mathbf{G}} \cdot \tilde{\mathbf{G}}^{-1}(\mathbf{y}_{t,\text{id}[j+1]})$$

with $\|\mathbf{f}_{t,j+1}\|_\infty \leq b/2$. By construction, for all $j \in [\ell]$, it holds that

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} \cdot \mathbf{u}_{t,\text{id}[j]} = \begin{pmatrix} \mathbf{y}_{t,\text{id}[j]} \\ -\mathbf{y}_{t,\text{id}[j+1]} + \mathbf{f}_{t,j+1} \end{pmatrix} \bmod q.$$

The decryptor computes

$$\begin{aligned} \tilde{\mu} &= d_t - \sum_{j \in [\ell]} \mathbf{c}_j^T \cdot \mathbf{u}_{t,\text{id}[j]} - \mathbf{c}_\ell^T \cdot \mathbf{x}_{t,\text{id}} \bmod q \\ &= \left\lfloor \frac{q}{2} \right\rfloor \cdot m_t + e_t \bmod q, \\ \text{and } \mu &= \left\lfloor \frac{2}{q} \cdot \left(\left\lfloor \frac{q}{2} \right\rfloor \cdot m_t + e_t \bmod q \right) \right\rfloor \bmod p \end{aligned}$$

for some noise term⁹

$$\begin{aligned} e_t &= \tilde{e}_t - \sum_{j=1}^{\ell-1} \mathbf{r}_j^T \cdot \mathbf{f}_{t,j} - \mathbf{r}_\ell^T \cdot (\mathbf{f}_{t,\ell} - \mathbf{z}_{t,\text{id}}) \\ &\quad - \sum_{j=0}^{\ell-1} \mathbf{e}_j^T \cdot \mathbf{u}_{t,\text{id}[j]} - \mathbf{e}_\ell^T \cdot \mathbf{x}_{t,\text{id}}. \end{aligned} \tag{A.1}$$

⁹Notice that the term $\mathbf{r}_0$ vanishes. Therefore, decryption would be correct even if the encryption algorithm picks $\mathbf{r}_0$ uniformly at random.

For decryption to be correct, it suffices to have

$$\left\| \left(\frac{2}{q} \cdot \left\lfloor \frac{q}{2} \right\rfloor - 1 \right) \cdot m_t + \frac{2}{q} \cdot e_t \right\|_\infty < \frac{1}{2}.$$

Since $\|m_t\|_\infty \leq 1$, the above is implied by

$$\|e_t\|_\infty < \frac{q}{4} - \frac{1}{2}. \quad (\text{A.2})$$

We next examine the norm of each component of e_t in Equation (A.1). From hypothesis, all of the below holds simultaneously except with probability at most ϵ :

$$\begin{aligned} \|\tilde{e}_t\|_\infty &\leq \sqrt{\varphi} \cdot \tilde{\sigma}, \\ \left\| \sum_{j=1}^{\ell-1} \mathbf{r}_j^T \cdot \mathbf{f}_{t,j} \right\|_\infty &\leq (\hat{\ell} - 1) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot \frac{b}{2}, \\ \left\| \mathbf{r}_\ell^T \cdot (\mathbf{f}_{t,\ell} - \mathbf{z}_{t,\text{id}}) \right\|_\infty &\leq 2 \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot \frac{b}{2}, \\ \left\| \sum_{j=0}^{\ell-1} \mathbf{e}_j^T \cdot \mathbf{u}_{t,\text{id}[j]} \right\|_\infty &\leq \hat{\ell} \cdot \varphi^{3/2} \cdot 3\ell_A n \cdot \sigma \cdot \frac{b}{2}, \\ \left\| \mathbf{e}_\ell^T \cdot \mathbf{x}_{t,\text{id}} \right\|_\infty &\leq \varphi^{3/2} \cdot n \cdot \sigma \cdot \frac{b}{2} \end{aligned}$$

where we applied the inequalities between 2-norm and ∞ -norm, as well as the Cauchy–Schwarz inequality. For example, except with probability $(\ell - 1)2^{-\varphi n}$, we have

$$\begin{aligned} \left\| \sum_{j=1}^{\ell-1} \mathbf{r}_j^T \cdot \mathbf{f}_{t,j} \right\|_\infty &\leq \sum_{j=1}^{\ell-1} \|\mathbf{r}_j^T \cdot \mathbf{f}_{t,j}\|_\infty \leq \sum_{j=1}^{\ell-1} \|\mathbf{r}_j^T \cdot \mathbf{f}_{t,j}\|_2 \\ &\leq \sum_{j=1}^{\ell-1} \sqrt{\varphi} \cdot \|\mathbf{r}_j\|_2 \cdot \|\mathbf{f}_{t,j}\|_2 \\ &\leq \sum_{j=1}^{\ell-1} \sqrt{\varphi} \cdot \|\mathbf{r}_j\|_2 \cdot \sqrt{\varphi n} \|\mathbf{f}_{t,j}\|_\infty \\ &\leq \sqrt{\varphi} \cdot (\hat{\ell} - 1) \cdot \sqrt{\varphi n} \sigma \cdot \sqrt{\varphi n} b / 2 \end{aligned}$$

where we first used triangle and norm inequalities, then the Cauchy–Schwarz inequality, where the ℓ_2 -expansion factor of $\mathcal{R}$ is upper bounded by $\sqrt{\varphi}$ and finally the inequalities $\|\mathbf{f}\|_2 \leq \sqrt{\varphi n} \cdot \|\mathbf{f}\|_\infty$ and $\|\mathbf{f}\|_\infty \leq b/2$, and $\Pr[\|\mathbf{r}^T\|_2 > \sqrt{(\hat{\ell} - 1)\varphi n \cdot \sigma}] \leq 2^{-\varphi n}$ by Lemma 3.3. The other terms are derived similarly.

Collecting all of the above error bounds, we have

$$\|e_t\|_\infty \leq ((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b/2 + \sqrt{\varphi} \cdot \tilde{\sigma} \quad (\text{A.3})$$

with probability at least ϵ/B . By the lower bound of q in Table 2, the inequality in (A.2) holds. By union bound over $t \in S \subseteq [B]$, the claim follows. $\square$

A.2. Proof of Corollary 5.3

Proof. Recall that an honestly generated ciphertext by BLE has the form

$$\begin{aligned} \mathbf{c}_j^T &= (\mathbf{r}_j, \mathbf{r}_{j+1})^T \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} + \mathbf{e}_j^T \bmod q \in \mathcal{R}_q^{3\ell_A n}, \\ \mathbf{c}_\ell^T &= \mathbf{r}_\ell^T \cdot \mathbf{B} + \mathbf{e}_\ell^T \bmod q \in \mathcal{R}_q^n, \\ \mathbf{d}^T &= \mathbf{r}_0^T \cdot \mathbf{Y}_{S,\epsilon} + \tilde{\mathbf{e}}^T + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{m}^T \bmod q \in \mathcal{R}_q^S \end{aligned}$$

The difference of ciphertext by BLE' can be expressed by error terms ξ_i and δ with $\|\xi_i\|_\infty, \|\delta\|_\infty \leq 2^D/2$, namely

$$\begin{aligned} \mathbf{c}'_i &= 2^D \lfloor \mathbf{c}_i / 2^D \rfloor = \mathbf{c}_i + \xi_i \\ \mathbf{d}' &= 2^D \lfloor \mathbf{d} / 2^D \rfloor = \mathbf{d} + \delta \end{aligned}$$

for $i = 0, \dots, \ell$. Thus, the resulting error term for $t \in S$ during decryption is increased by

$$e'_t = \delta - \sum_{j \in [\ell]} \xi_j^T \mathbf{u}_{t, \text{id}[:, j]} - \xi_\ell^T \mathbf{x}_{t, \text{id}}$$

and we find

$$\begin{aligned} \|e'\|_\infty &\leq 2^D/2 + \hat{\ell} \cdot 2^D/2 \cdot \varphi \cdot b/2 + 2^D/2 \cdot \varphi \cdot b/2 \\ &= 2^{D-1} \cdot (1 + (\hat{\ell} + 1) \cdot \varphi \cdot b/2) \end{aligned}$$

where we used $h = 1$, $\|\mathbf{x}\|_\infty \leq b/2$ from the parameters and φ comes from the ℓ_∞ -expansion factor of $\mathcal{R}$. This increase in the error is compensated by and increased bound on q , namely, $\|e_t + e'_t\|_\infty \leq q/4 - 1/2$ holds if

$$q \geq 4\|e_t + e'_t\|_\infty + 2$$

and we have that

$$\begin{aligned} &4\|e_t + e'_t\|_\infty + 2 \\ &\leq 4\|e_t\|_\infty + 2 + \|e'_t\|_\infty \\ &= \left(4((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b/2 + 4 \cdot \sqrt{\varphi} \tilde{\sigma} + 2 \right) \\ &\quad + 4 \cdot \left(2^{D-1} \cdot (1 + (\hat{\ell} + 1) \cdot \varphi \cdot b) \right) \\ &= 4 \cdot \left(((3\ell_A + 1)\hat{\ell} + 2) \cdot \varphi^{3/2} \cdot n \cdot \sigma \cdot b/2 + \sqrt{\varphi} \tilde{\sigma} \right) \\ &\quad + 4 \cdot 2^{D-1} \cdot (1 + (\hat{\ell} + 1) \cdot \varphi \cdot b/2) + 2 \end{aligned}$$

from the bound on e_t in Theorem 5.2 and the bound on e'_t from above. The claim follows. $\square$

A.3. Proof of Theorem 5.5

A challenge in the proof is handling the sub-registries S , since some of them may not have a key registered for id. However, we also can not use a straightforward hybrid argument of the $t \in S$, because the shared randomness intertwines them in $\mathbf{d}$. To handle this formally, we need a technical argument. The underlying idea is to “formally extend” the values in a registry $\text{reg}[t, \text{id}]$ by zeroes, i.e. instead of making many case distinctions for handling the case $\text{reg}[t, \text{id}[:, j]] = \perp$, we pretend it contains $\mathbf{0}$. As we see in the proof, this does not affect the linear algebra (as everything just continues to be zeroes), but we can specify and analyse the proof without extensive case distinctions.

Proof. Recall that in the $\text{IND\$-CPA}_{\text{BLE}, \mathcal{A}}^0(1^\lambda)$ experiment, after $(\text{pp}, \text{st}, \text{reg}) \leftarrow \text{Setup}(1^\lambda)$ generates the public parameters, the adversary gets access to oracles $\text{KGen}\mathcal{O}$ and $\text{Upd}\mathcal{O}$. Let $S^*, \text{id}^*, (\text{msg}_t)_{t \in S^*}$ be the adversary’s challenge query and the challenge be $\text{ctxt}^* \leftarrow \text{Enc}(\text{pp}, \text{st}, S^*, \text{id}^*, (\text{msg}_t^*)_{t \in S^*})$. Without loss of generality, we assume that $\text{msg}_t^* = 0$ for all $t \in S^*$, since ciphertexts encrypting non-zero messages can be obtained by simply by adding $\lfloor \frac{q}{p} \rfloor \cdot \text{msg}^*$ to the $\mathbf{d}$ component. Let $\mathbf{r}_j$, $\mathbf{e}_j$, $\tilde{\mathbf{e}}$, $\mathbf{c}_i$, $\mathbf{d}$, etc. be as in Enc . Let reg denote the registry at the time of the challenge ciphertext request. In the following, we drop the superscript $*$ from all challenge ciphertext variables for convenience, in particular, $\text{id} = \text{id}^*$ and $S = S^*$. Let $\ell = |\text{id}| \leq \hat{\ell}$. We define a sequence of games $G_{-4}, \dots, G_0, \dots, G_\ell$ where G_{-4} and G_ℓ are the $\text{IND\$-CPA}_{\text{BLE}, \mathcal{A}}^0(1^\lambda)$ and $\text{IND\$-CPA}_{\text{BLE}, \mathcal{A}}^1(1^\lambda)$ experiments defined in Theorem 5.5, respectively. The intermediate games are defined below.

Game G_{-3} : This game is identical to G_{-4} except that $\text{KGen}\mathcal{O}$ samples public keys pk uniformly at random from $\mathcal{R}_q^n$. This change is well-defined since the corresponding secret keys sk are never used in G_{-4} . Via a reduction to $(Q_{\text{KGen}}$ instances of) LWE , we immediately have

$$|\Pr[G_{-4} = 1] - \Pr[G_{-3} = 1]| \leq Q_{\text{KGen}} \cdot \text{Adv-LWE}_{\mathcal{A}}^{\text{lwe}}(\lambda).$$

Game G_{-2} : This game is identical to G_{-3} except that we change how the challenge ciphertext component $\mathbf{d}$ is derived, in a functionally equivalent way. To explain this change, we first need to introduce some notation.

Recall that the challenge ciphertext takes the form

$$\begin{aligned}\mathbf{c}_j^T &= \begin{pmatrix} \mathbf{r}_j^T & \mathbf{r}_{j+1}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} + \mathbf{e}_j^T \bmod q \in \mathcal{R}_q^{km_A}, \\ \mathbf{c}_\ell^T &= \mathbf{r}_\ell \cdot \mathbf{B} + \mathbf{e}_\ell^T \bmod q \in \mathcal{R}_q^{m_B}, \\ \mathbf{d}^T &= \mathbf{r}_0^T \cdot \mathbf{Y}_{S,\epsilon} + \tilde{\mathbf{e}}^T \bmod q \in \mathcal{R}_q^S.\end{aligned}$$

We partition S into two sets, $S = S_0 \cup S_0^\dagger$, according to whether $\text{reg}[t, \epsilon]$ is a terminator or not. Namely, we assign $t \in S$ to either S_0 or $S_0^\dagger$ as follows:

- (1) $t \in S_0$ if $\text{reg}[t, \epsilon] \neq \text{term}[t]$.
- (2) $t \in S_0^\dagger$ if $\text{reg}[t, \epsilon] = \text{term}[t]$ (i.e., uniform over $\mathcal{R}_q^n$).

Due to how reg is structured, if $t \in S_0$ we must have that the node $\text{reg}[t, \epsilon]$ has k children. Inductively, we can partition $S_j = S_{j+1} \cup S_{j+1}^\dagger$ for $j = 0, \dots, \ell - 1$ via

- (1) $t \in S_j$ if $\text{reg}[t, \text{id}[:j]] \neq \text{term}[t]$.
- (2) $t \in S_j^\dagger$ if $\text{reg}[t, \text{id}[:j]] = \text{term}[t]$ (i.e., uniform over $\mathcal{R}_q^n$).

Note that $S_j^\dagger$ contains exactly those $t \in S$ which terminate (with terminator $\text{term}[t]$) at depth/length j . By construction,

$$S_j = S_{j+1} \uplus S_{j+1}^\dagger = \text{ and } S = S_\ell \uplus S_\ell^\dagger \uplus \dots \uplus S_0^\dagger$$

are disjoint unions. For notational convenience, let us write

$$\mathbf{y}_{t, \text{id}[:j]} := \text{reg}[t, \text{id}[:j]]$$

and define the matrices $\mathbf{Y}_{S, \text{id}[:j]}^*$, $\mathbf{Y}_{S, \text{id}[:j]}^\dagger \in \mathcal{R}_q^{n \times S}$ whose t -th columns are given by

$$(\mathbf{y}_{t, \text{id}[:j]}^*, \mathbf{y}_{t, \text{id}[:j]}^\dagger) = \begin{cases} (\mathbf{y}_{t, \text{id}[:j]}, \mathbf{0}) & t \in S_j \\ (\mathbf{0}, \mathbf{y}_{t, \text{id}[:j]}) & t \in S_j^\dagger \\ (\mathbf{0}, \mathbf{0}) & t \notin S_j \cup S_j^\dagger \end{cases}$$

respectively. That is, non-zero columns of $\mathbf{Y}_{S, \text{id}[:j]}^\dagger$ correspond exactly to those sub-registries which terminate at depth j (with $\mathbf{y}_{t, \text{id}[:j]} = \text{term}[t] = \text{reg}[t, j]$). Non-zero columns of $\mathbf{Y}_{S, \text{id}[:j]}^*$ correspond exactly to continuing paths (with $\mathbf{y}_{t, \text{id}[:j]} = \text{reg}[t, \text{id}[:j]]$). Define

$$\mathbf{Y}_{S, \text{id}[:j]} = \mathbf{Y}_{S, \text{id}[:j]}^* + \mathbf{Y}_{S, \text{id}[:j]}^\dagger$$

where the only non-zero columns are indexed by S_j and $S_j^\dagger$.

Correspondingly, define the matrices $\mathbf{U}_{S, \text{id}[:j]}^* \in \mathcal{R}^{km_A \times S}$ whose t -th columns are given by

$$\mathbf{u}_{t, \text{id}[:j]}^* = \begin{cases} \mathbf{u}_{t, \text{id}[:j]} = \left(-\tilde{\mathbf{G}}^{-1} (\mathbf{y}_{t, \text{id}[:j]} \| \mathbf{i}) \right)_{i \in [k]} & t \in S_j \\ \mathbf{0} & t \notin S_j. \end{cases}$$

By definition, $\mathbf{U}_{S, \text{id}[:j]}$ contains the approximate preimages to the (non-zero) columns of $\mathbf{Y}_{S, \text{id}[:j+1]}$ which are part of the membership witness.¹⁰ In particular, the following holds modulo q :

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} \cdot \mathbf{U}_{S, \text{id}[:j]}^* = \begin{pmatrix} \mathbf{Y}_{S, \text{id}[:j]} - \mathbf{Y}_{S, \text{id}[:j]}^\dagger \\ -\mathbf{Y}_{S, \text{id}[:j+1]} + \mathbf{F}_{S, \text{id}[:j+1]} \end{pmatrix}, \quad (\text{A.4})$$

where $\mathbf{F}_{S, \text{id}[:j+1]} = \mathbf{Y}_{S, \text{id}[:j+1]} - \tilde{\mathbf{G}} \cdot \tilde{\mathbf{G}}^{-1}(\mathbf{Y}_{S, \text{id}[:j+1]})$ has

$$\|\mathbf{F}_{S, \text{id}[:j+1]}\|_\infty \leq b^h/2 = b/2 \text{ for } j \in [\ell].$$

Note that for $j \in [\ell]$, $\mathbf{F}_{S, \text{id}[:j+1]}$ has non-zero columns only where $\mathbf{Y}_{S, \text{id}[:j+1]}$ has, that is, only in $S_j = S_{j+1} \uplus S_{j+1}^\dagger$. Combining Equation (A.4) with the fact that

$$\mathbf{c}_j^T = \begin{pmatrix} \mathbf{r}_j^T & \mathbf{r}_{j+1}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}} \end{pmatrix} + \mathbf{e}_j^T \bmod q$$

¹⁰Strictly speaking, preimages to all k string $\mathbf{y}_{t, \text{id}[:j]} \| \mathbf{i}$ for $i \in [k]$ are contained in $\mathbf{u}_{t, \text{id}[:j]}^*$. The definition of $\mathbf{A}_{\text{id}}$, specifically $\mathbf{i}_{\text{id}_j}^T \otimes \tilde{\mathbf{G}}$ “chooses” the relevant preimage.

we obtain

$$\begin{aligned} \mathbf{c}_j^\top \cdot \mathbf{U}_{S,\text{id}[j]}^* &= \begin{pmatrix} \mathbf{r}_j^\top & \mathbf{r}_{j+1}^\top \end{pmatrix} \begin{pmatrix} \mathbf{Y}_{S,\text{id}[j]} - \mathbf{Y}_{S,\text{id}[j]}^\dagger \\ -\mathbf{Y}_{S,\text{id}[j+1]} + \mathbf{F}_{S,\text{id}[j+1]} \end{pmatrix} \\ &\quad + \mathbf{e}_j^\top \cdot \mathbf{U}_{S,\text{id}[j]}^* \bmod q. \end{aligned}$$

Summing both sides over $j \in [\ell]$ and adding an additional term of $\mathbf{r}_\ell^\top \mathbf{Y}_{S,\text{id}}$, we obtain

$$\begin{aligned} &\sum_{j=0}^{\ell-1} \mathbf{c}_j^\top \cdot \mathbf{U}_{S,\text{id}[j]}^* + \mathbf{r}_\ell^\top \mathbf{Y}_{S,\text{id}} \\ &= \mathbf{r}_0^\top \mathbf{Y}_{S,\epsilon} - \sum_{j=0}^{\ell-1} \mathbf{r}_j^\top \mathbf{Y}_{S,\text{id}[j]}^\dagger + \sum_{j=0}^{\ell} \begin{pmatrix} \mathbf{r}_j^\top & \mathbf{e}_j^\top \end{pmatrix} \mathbf{L}_j \end{aligned}$$

where the leakage matrices $\mathbf{L}_j$ are defined as

$$\mathbf{L}_j = \begin{cases} \begin{pmatrix} \mathbf{0} \\ \mathbf{U}_{S,\epsilon}^* \end{pmatrix} & j = 0 \quad / \text{ zero at column } t \notin S_0 \\ \begin{pmatrix} \mathbf{F}_{S,\text{id}[j]} \\ \mathbf{U}_{S,\text{id}[j]}^* \end{pmatrix} & 0 < j < \ell \quad / \text{ zero at column } t \notin S_{j-1} \\ \begin{pmatrix} \mathbf{F}_{S,\text{id}[j]} \\ \mathbf{0} \end{pmatrix} & j = \ell. \quad / \text{ zero at column } t \notin S_{\ell-1} \end{cases}$$

Consequently, we can rewrite $\mathbf{d} = \mathbf{r}_0^\top \cdot \mathbf{Y}_{S,\epsilon} + \tilde{\mathbf{e}}^\top \bmod q$ as

$$\begin{aligned} \mathbf{d}^\top &= \sum_{j=0}^{\ell-1} \left(\mathbf{c}_j^\top \cdot \mathbf{U}_{S,\text{id}[j]}^* + \mathbf{r}_j^\top \mathbf{Y}_{S,\text{id}[j]}^\dagger \right) + \mathbf{r}_\ell^\top \mathbf{Y}_{S,\text{id}} \\ &\quad - \sum_{j=0}^{\ell} \begin{pmatrix} \mathbf{r}_j^\top & \mathbf{e}_j^\top \end{pmatrix} \mathbf{L}_j + \tilde{\mathbf{e}}^\top \bmod q. \end{aligned}$$

Overall, the change between games $\mathbf{G}_{-3}$ and $\mathbf{G}_{-2}$ is merely conceptual, i.e. a different way to compute $\mathbf{d}$. Thus $\Pr[\mathbf{G}_{-3} = 1] = \Pr[\mathbf{G}_{-2} = 1]$.

Game $\mathbf{G}_{-1}$: In this game, we substitute the error $\tilde{\mathbf{e}}$ in $\mathbf{d}$ by a sum

$$\tilde{\mathbf{e}} \approx \sum_{j=0}^{\ell} (\mathbf{e}_j^\dagger + \bar{\mathbf{e}}_j)$$

where

- For $j \in [\ell]$, $\mathbf{e}_j^\dagger$ and $\bar{\mathbf{e}}_j$ are zero everywhere except for the entries $t \in S_j^\dagger$ which are sampled from χ_e and $\bar{\chi}_e$, respectively;
- $\mathbf{e}_\ell^\dagger$ and $\bar{\mathbf{e}}_\ell$ are zero everywhere except for the entries $t \in S_{\ell-1} = S_\ell \uplus S_\ell^\dagger$ which are sampled from χ_e and $\bar{\chi}_e$, respectively.

Looking ahead, we need $\mathbf{e}_j^\dagger$ and $\bar{\mathbf{e}}_j$ for an LLWE application to the j -th ciphertext $\mathbf{c}_j$ (i.e. at depth j), hence, we require them to be non-zero exactly in the components of $S_j^\dagger$ (except for $\mathbf{c}_\ell$, where LLWE is applied to terminators *and* the (random) public keys, hence $S_\ell \uplus S_\ell^\dagger = S_{\ell-1}$).

Substituting $\tilde{\mathbf{e}}$ in the expression of $\mathbf{d}$, we have

$$\begin{aligned} \mathbf{d}^\top &= \sum_{j=0}^{\ell-1} \left(\mathbf{c}_j^\top \cdot \mathbf{U}_{S,\text{id}[j]}^* + \mathbf{r}_j^\top \mathbf{Y}_{S,\text{id}[j]}^\dagger + (\mathbf{e}_j^\dagger)^\top \right) \\ &\quad + \mathbf{r}_\ell^\top \mathbf{Y}_{S,\text{id}} + (\mathbf{e}_\ell^\dagger)^\top - \sum_{j=0}^{\ell} \left(\begin{pmatrix} \mathbf{r}_j^\top & \mathbf{e}_j^\top \end{pmatrix} \mathbf{L}_j + \bar{\mathbf{e}}_j \right) \bmod q. \end{aligned}$$

By assumption, the sum $\chi_e + \bar{\chi}_e$ is statistically δ -close to $\tilde{\chi}_e$, and the above method of generating $\mathbf{d}$ is otherwise identical to $\mathbf{G}_{-2}$. After a union bound over the $|S| \leq B$ components of $\tilde{\mathbf{e}} \in \mathcal{R}_q^S$, we therefore find

$$|\Pr[\mathbf{G}_{-2} = 1] - \Pr[\mathbf{G}_{-1} = 1]| \leq B \cdot \delta = \text{negl}(\lambda).$$

Game G_i for $i \in [\ell]$: We replace $\mathbf{c}_i$ and the non-trivial entries (namely for $t \in S_j^\dagger$) of $\mathbf{r}_j^\top \mathbf{Y}_{S, \text{id}[:, j]}^\dagger + (\mathbf{e}_j^\dagger)^\top \bmod q$ with uniformly random vectors over $\mathcal{R}_q$. Observe that in G_{i-1} the dependency of $\mathbf{d}$ on the secrets $(\mathbf{r}_i, \mathbf{e}_i, \mathbf{e}_i^\dagger)$ is only via $\mathbf{c}_i$, $\mathbf{r}_i^\top \mathbf{Y}_{S, \text{id}[:, i]}^\dagger + (\mathbf{e}_i^\dagger)^\top \bmod q$ (which has $|S_i^\dagger|$ non-zero entries), and $(\mathbf{r}_i^\top, \mathbf{e}_i^\top) \cdot \mathbf{L}_i + \bar{\mathbf{e}}_i$. This change can therefore be justified via a reduction from the Leaky LWE assumption. More explicitly, the LWE instance is given by

$$\mathbf{r}_i^\top (\mathbf{A}, \mathbf{Y}_{S, \text{id}[:, i]}^\dagger) + (\mathbf{e}_i^\top, (\mathbf{e}_i^\dagger)^\top) = (\mathbf{c}_i, \mathbf{r}_i^\top \mathbf{Y}_{S, \text{id}[:, i]}^\dagger + (\mathbf{e}_i^\dagger)^\top)$$

and the noisy leakage is $(\mathbf{r}_i^\top, \mathbf{e}_i^\top) \mathbf{L}_i + \bar{\mathbf{e}}_i$. Thus, we find that

$$|\Pr[G_{i-1} = 1] - \Pr[G_i = 1]| \leq \text{Adv-LLWE}_{\mathcal{A}}^{\text{lwe}}(\lambda).$$

Game G_ℓ : We replace $\mathbf{c}_\ell$ and the non-trivial entries (namely for $t \in S_{\ell-1}$) of $\mathbf{r}_\ell^\top \mathbf{Y}_{S, \text{id}} + (\mathbf{e}_\ell^\dagger)^\top \bmod q$ with uniformly random vectors over $\mathcal{R}_q$. Observe that in $G_{\ell-1}$ the dependency of $\mathbf{d}$ on the secrets $(\mathbf{r}_\ell, \mathbf{e}_\ell, \mathbf{e}_\ell^\dagger)$ is only via $\mathbf{c}_\ell$, $\mathbf{r}_\ell^\top \mathbf{Y}_{S, \text{id}} + (\mathbf{e}_\ell^\dagger)^\top \bmod q$ (which has $|S_{\ell-1}|$ non-zero entries), and $(\mathbf{r}_\ell^\top, \mathbf{e}_\ell^\top) \cdot \mathbf{L}_\ell + \bar{\mathbf{e}}_\ell$. This change can therefore be justified via a reduction from the Leaky LWE assumption. Formally, we have

$$|\Pr[G_{i-1} = 1] - \Pr[G_i = 1]| \leq \text{Adv-LLWE}_{\mathcal{A}}^{\text{lwe}}(\lambda).$$

Note that in G_ℓ all of $\mathbf{c}_0, \dots, \mathbf{c}_\ell$ are uniformly random. Furthermore, over the course of G_0 to G_ℓ , we also replaced the non-zero entries of $\mathbf{r}_j^\top \mathbf{Y}_{S, \text{id}[:, j]}^\dagger + (\mathbf{e}_j^\dagger)^\top \bmod q$ and $\mathbf{r}_\ell^\top \mathbf{Y}_{S, \text{id}} + (\mathbf{e}_\ell^\dagger)^\top \bmod q$ with uniformly random vectors over $\mathcal{R}_q$. Since the indices of the non-zero entries of these vectors partition S , we conclude that all entries of $\mathbf{d}$ are uniformly random in G_ℓ . Therefore, $G_\ell = \text{IND\$-CPA}_{\text{BLE}, \mathcal{A}}^1(1^\lambda)$. $\square$