Poster: Scalable Registration-Based Encryption from Lattices

Michael Klooß
KIT & KASTEL, Germany
klooss@mail.informatik.kit.edu

Russell W. F. Lai
Aalto University, Finland
russell.lai@aalto.fi

Jan Niklas Siemer
King's College London, UK
jan.siemer@kcl.ac.uk

Monisha Swarnakar
Aalto University, Finland
monisha.swarnakar@aalto.fi

***Abstract*—Registration-Based Encryption (RBE) is a public-key encryption mechanism which allows users 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 and Motivation

In today's highly digitized world, most encrypted communication utilises Public-Key Encryption (PKE) or a specialisation of this primitive such as Key Exchange Mechanisms (KEMs). These primitives require knowledge of the receiver's public key to encrypt a message. The high volume of communication and the large number of conversation partners in our modern society quickly evolves into a complex key-management problem. While key management is handled well by modern web browsers for TLS connections, establishing a common ground for applications like end-to-end-encrypted email has proven difficult.

A potential solution to the key-management problem was proposed in 1984 by Adi Shamir. He introduced a primitive called Identity-Based Encryption (IBE) [1], which encrypts with respect to an identity string – a format suited to be memorised by humans – and the public key of a trusted third entity, e.g. an organisation that the receiver belongs to. This approach drastically reduces the number of public keys handled by a single entity. Additionally, typical and memorisable formats for URLs `identity.organisation` and emails `identity@organisation` suit the combination of these two pieces of information well. However, recipients obtain their private decryption keys directly from the trusted third party, which introduces the "key-escrow" problem: the trusted third party can generate any identity's private key and therefore, decrypt any message encrypted with respect to its public key. Thus, IBE is an unsuitable candidate to enable end-to-end-encrypted email systems.

Apart from threshold IBE, Registration-Based Encryption (RBE) [2] yields a solution to the key escrow problem by replacing the trusted third entity with a publicly auditable database, called "key curator". In RBE, users generate their own key pairs and register their public keys with the key curator, who computes a small digest of all registered public keys as its public key and provides a membership witness to registered users. This allows users to decrypt using their own private key and membership witness, while encryptors only need the recipient's identity and the key curator's public key.

The first efficient RBE was proposed in 2022 [3], but its construction suffered from critical shortcomings: supporting only a polynomial number of identities and lacking post-quantum security by relying on pairings. However, their concept of building an RBE from a vector commitment and a public-key encryption scheme inspired the first plausibly post-quantum RBE construction from lattice assumptions [4]. Their construction was derived from a closely related primitive called Laconic Encryption (LE). While their construction is theoretically sound, their system yields highly impractical ciphertext sizes of 7.2 GB for large organisations with up to $N = 2^{30}$ users and arbitrary identity strings.

2. The Efficiency Bottleneck

While RBE requires the number of membership witness updates to be bounded by $\mathcal{O}(\log N)$ for any registered identity, LE does not raise this requirement. Otherwise, the primitives are mostly identical.

The generic transform from LE to RBE [4] utilises B LE instances for a maximum number of registered RBE users $N = 2^B$. In brief, users' public keys are stored in LE instances whose sizes are progressively larger powers of two. A new public key is inserted into the first LE instance without any users. If two LE instances are of the same size, they are merged into a larger LE instance so that all LE instances are of size distinct powers of two. Every merge operation results in one of the LE instances containing twice the number of users and one returning to an empty state. This leads to a maximum of $\log N = B$ merge operations that a single identity's public key encounters, which is the only operation required to update the membership witness. This behaviour satisfies RBE's efficiency requirement but requires the sender to encrypt with respect to all B LE instances simultaneously. Thus, the ciphertext size of the RBE grows multiplicatively with B, i.e. $|\mathsf{ctxt}_{\mathsf{RBE}}| = B \cdot |\mathsf{ctxt}_{\mathsf{LE}}|$.

3. Our Solution: Batched Laconic Encryption

To reduce the ciphertext size and growth factor, we introduce **Batched Laconic Encryption (BLE)**. Our key observation is that the B parallel LE instances in an RBE transformation are not unrelated – they all encrypt to the *same* identity and can share significant structural components.

Instead of generating independent ciphertexts, BLE enables encrypting to a batch of B instances simultaneously with an *additive* rather than multiplicative overhead. Our BLE construction utilises the fact that significant parts of the ciphertext are determined by the vector commitment to commit to the identity of the recipient. As this identity is the same across all B LE instances in the previously described transform, we generate and send this part of the ciphertext only once. The main technical difficulty of our approach is sharing the randomness used to generate the commitment across all public-key encryption instances. Thus, for a laconic encryption scheme whose ciphertext contains a vector commitment (VC) part and a PKE-related part $\mathsf{ctxt}_{\mathsf{LE}} = \left(\mathsf{ctxt}^{\mathsf{VC}}_{\mathsf{LE}}, \mathsf{ctxt}^{\mathsf{PKE}}_{\mathsf{LE}}\right)$, our BLE achieves a ciphertext size of $|\mathsf{ctxt}_{\mathsf{BLE}}| = |\mathsf{ctxt}^{\mathsf{VC}}_{\mathsf{LE}}| + B \cdot \mathsf{ctxt}^{\mathsf{PKE}}_{\mathsf{LE}}$. As $|\mathsf{ctxt}^{\mathsf{VC}}_{\mathsf{LE}}| \gg \mathsf{ctxt}^{\mathsf{PKE}}_{\mathsf{LE}}$, this yields a significant improvement compared to the generic LE-to-RBE transform.

3.1. Further Optimisations

Beyond the architectural shift to BLE, we employ several further optimisations to reduce key and ciphertext sizes.

- **Approximate Gadgets:** We replace exact gadget matrices with an approximate gadget matrices, allowing us to drop low-order bits and significantly reduce the dimension of the matrix $\mathbf{A}$.
- **Optimal Tree-Arity:** Our construction builds on a Merkle tree-based vector commitment, which makes the ciphertext size scale linearly with $k/\log k$ with k denoting the arity of the Merkle tree. We minimize this by choosing $k = 3$ (ternary trees) instead of binary trees.
- **Computational Reductions:** We replace statistical arguments (Leftover Hash Lemma) with a tighter computational reduction to the Learning with Errors (LWE) assumption, yielding smaller parameters.

4. Results

Our resulting RBE scheme is the first post-quantum construction that simultaneously supports a large number of users and an arbitrary identity space while remaining practically sized. For up to 2^{30} registered users at 128-bit security, our scheme achieves a ciphertext size of 7.0 MB, improving on prior state-of-the-art constructions by a factor of $1,000$.

We verified these results via an open-source prototype implementation in Rust. Utilising hardware-accelerated NTT and high-throughput discrete Gaussian samplers, our benchmarks demonstrate encryption and decryption execution times of 15.56 ms and 10.43 ms respectively.

5. Future Work

While we have achieved single-digit MB ciphertexts, pushing this down to the kilobyte range likely requires replacing the underlying Merkle tree with a more succinct vector commitment. Doing so while retaining a transparent setup remains a challenging open problem for future research.

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 (cit. on p. 1).

[2] 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 (cit. on p. 1).

[3] 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 (cit. on p. 1).

[4] 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 (cit. on pp. 1, 2).

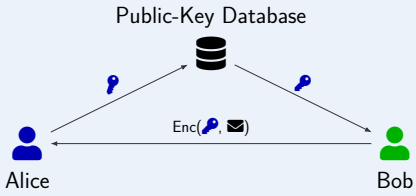
Registration-Based Encryption

Scalable and From Standard Lattice Assumptions

King's College London · Aalto University · Karlsruher Institut für Technologie · KASTEL

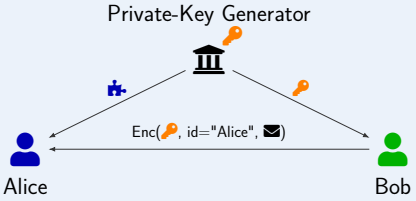
Primitives to Send an Encrypted Email

PKE



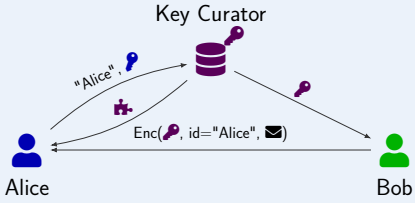
+ Lightweight
– Key management
– Recipient exposed on key retrieval

IBE



+ Key management
– Trusted PKG

RBE



+ Key management
+ No trusted authority
– Update frequency

Primitives: LE, BLE & RBE

Laconic Encryption

- $(\mathsf{pp}, \mathsf{st}, \mathsf{reg}) \leftarrow \mathsf{Setup}(1^n)$: generates initial state 🔑 and registry 🗄
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\mathsf{pp})$: generates a key-pair with pk 🔑
- $\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{st}, \mathsf{id}, \mathsf{msg})$: encrypts msg for identity id
- $\mathsf{msg} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{wit}, \mathsf{ctxt})$
- $\mathsf{st}' \leftarrow \mathsf{Upd}^{\mathsf{reg}}(\mathsf{pp}, \mathsf{st}, \mathsf{id}, \mathsf{pk})$: updates state 🔑 and registry 🗄 upon registration of a new identity id according to its pk 🔑
- $\mathsf{wit} \leftarrow \mathsf{WGen}^{\mathsf{reg}}(\mathsf{pp}, \mathsf{st}, \mathsf{id})$: generates 🧩 called witness

RBE — *LE + Efficiency Requirement:* Each identity's witness wit 🧩 updates at most $O(\log N)$ times; N defines the maximum number of registered identites.

Batched LE — *LE* with reg containing B sub-registries + *Efficiency Requirement:* only *small parts* of |ctxt| allowed to scale linearly with B

Transforming LE to RBE

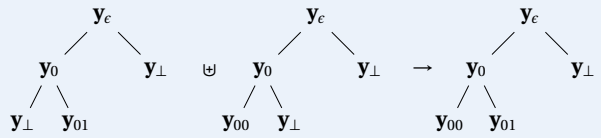
Encryption against B LE instances ⇒ |ctxt| scales **linearly** with B

Take B LE instances for a max. number of users $N = 2^B$. On registry of a new user, add it to an empty LE. Iteratively merge two LEs if there are two LE instances with the same number of users, leaving one empty.
⇒ Max. number of merges (witness updates) per identity $\leq \log N$

BLE to RBE — Sub-registries of BLE take over behaviour of multiple LE instances and their merging behaviour.

DKLLMR Construction [1]

Vector Commitment: SIS-based Merkle-Tree with $\mathsf{H}(\mathbf{x}) = \mathbf{A}_0 \| \mathbf{A}_1 \cdot \mathbf{G}^{-1}(\mathbf{x})$, public keys $\mathbf{y}_{00}, \mathbf{y}_{01}$, terminator symbol $\mathbf{y}_\perp$ and merge operation $\uplus$



PKE: Dual Regev Encryption w.r.t. $(\mathbf{A}_{\mathsf{id}}, \mathbf{v})$

$\mathsf{wit}_{\mathsf{id}} = -\mathbf{G}^{-1}\left(\left(\mathbf{y}_{\mathsf{id}_{|i:|}\|b}\right)_{b\in\{0,1\}}\right)_{i\in[\ell]}$
$\Rightarrow \mathbf{A}_{\mathsf{id}} \cdot \mathsf{wit}_{\mathsf{id}} = \mathbf{v}$

$$\mathbf{A}_{\mathsf{id}} = \begin{pmatrix} \mathbf{A}_0\|\mathbf{A}_1 & & \\ \mathbf{i}_{\mathsf{id}_1}\otimes\mathbf{G} & \ddots & \\ & \ddots & \mathbf{A}_0\|\mathbf{A}_1 \\ & & \mathbf{i}_{\mathsf{id}_\ell}\otimes\mathbf{G} \quad \mathbf{B} \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} \mathbf{y}_\epsilon \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix},$$

where $\|$ concatenates matrices horizontally and $\mathbf{i}_j$ is the j-th unit vector, i.e.

$$\mathsf{ctxt} = (\mathbf{c}^T, d), \quad \mathbf{c}^T \approx \mathbf{r}^T\mathbf{A}_{\mathsf{id}} \bmod q, \quad d \approx \mathbf{r}^T\mathbf{v} + \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu \bmod q.$$

Our Optimisations

1. BLE reuses $\mathbf{c}^T$ to share randomness along path to id across all B trees
2. Approximate $\mathbf{G}$ with optimal base $b \approx q^{1/3}$ to reduce |columns| of $\mathbf{A}$
3. Replace statistical arguments in game hops by computational ones
4. Lossy compression of ciphertext

Comparison to Prior Work in Practice

Open-source implementation in Rust supporting hardware-accelerated NTT and discrete Gaussian sampler with high throughput.

	Time (ms)				Size (MB)	
	Upd	Enc	WGen	Dec	ctxt	st
[1]	5 · 1565	150 · 2.95	5 · 0.061	5 · 6.27	150 · 49	8
Our RBE	33.24	15.56	3.38	10.43	7.0	0.34

Fig. 1: Benchmarks on laptop for $N = 2^{30}$, $|ID| \geq 2^{256}$, and $n = 1{,}000$. LE of [1] scaled to RBE.

Conclusion

- Reduction of ciphertext size from 7.2 **G**B [1] to 7.0 **M**B
- Highly scalable: $N = 2^{128}$ users results in |ctxt| = 7.1 MB
- Use-cases: Low frequency, high bandwidth (e.g. emails in companies)
- Further improvements will require different vector commitment

References & Further Details

[1] 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. DOI: 10.1007/978-3-031-30620-4_14

- Paper: https://jnsiemer.de/scalable_rbe
- Implementation: jnsiemer/scalable_rbe_prototype

Michael Klooß, Russell W. F. Lai, Jan Niklas Siemer & Monisha Swarnakar
Karlsruhe Institute of Technology & KASTEL | Aalto University | King's College London