DoS-resistant Oblivious Message Retrieval and Snake-eye Resistant PKE from LWE

Zeyu Liu[1] Katerina Sotiraki[1] Eran Tromer[2] Yunhao Wang[1]

[1]Yale University
[2]Boston University

June 8, 2024

Abstract

Oblivious message retrieval (OMR) allows messages resource-limited recipients to outsource the message retrieval process without revealing which messages are pertinent to which recipient. Its realizations in recent works leave an open problem: can an OMR scheme be both practical and provably secure against spamming attacks from malicious senders (i.e., DoS-resistant) under standard assumptions?

In this paper, we first prove that a prior construction OMRp2 is DoS-resistant under a standard LWE assumption, resolving an open conjecture of prior works. Then, we present DoS-PerfOMR: a provably DoS-resistant OMR construction that is 12x faster than OMRp2, and (almost) matches the performance of the state-of-the-art OMR scheme that is *not* DoS-resistant (proven by the attacks we show).

As a building block, we analyze the *snake-eye resistance* property for general PKE schemes. We construct a new lattice-based PKE scheme, LWEmongrass that is provably snake-eye resistant and has better efficiency than the PVW scheme underlying OMRp2. We also show that the natural candidates (e.g., RingLWE PKE) are not snake-eye resistant.

Of independent interest, we introduce two variants of LWE with side information, as components towards proving the properties of LWEmongrass, and reduce standard LWE to them for the parameters of interest.

Contents

1 Introduction

Metadata privacy is gaining increasing attention, as a crucial complement to end-to-end encryption of message contents. Metadata can leak sensitive information via traffic analysis and deductions against auxiliary information, and thus protecting metadata is essential for applications such as anonymous messaging [54, 21, 37, 10, 7], privacy preserving analytics [10], and privacy preserving cryptocurrencies [42, 8, 14].

A particularly challenging component of metadata privacy is recipient privacy: no one should know who is the recipient of a specific message, except for the sender and recipient. And yet, recipients wish to efficiently retrieve their messages without downloading and scanning all the messages carried by the communication system.

To address this issue, recent works introduced cryptographic schemes that let the recipient outsource the job to an untrusted server. Fuzzy Message Detection [7] proposes a decoy-based scheme, where the server sees many false positives alongside the pertinent messages forwarded to the recipient. Although these decoys reduce the information learned by the server, the set of possible pertinent messages is still narrowed down, resulting in a weak security guarantee and possible attacks [51]. Then, the follow-up works [34, 39] independently introduce two similar primitives, that achieve stronger privacy: Oblivious Message Retrieval (OMR) and Private Signaling (PS) respectively. We focus on OMR, which is stronger than PS in terms of both functionality and privacy guarantees (see discussion in Section 2).

While the existing OMR constructions [34, 35, 36] all achieve full privacy, there is one major open issue: *spamming* in the malicious model (the so-called *Denial-of-Service (DoS)* model [34, Sec 8]). In more detail, in the DoS-model, a malicious sender can craft a malicious message deemed as *pertinent* by numerous honest recipients (not just a single one, which is of course always possible even if the sender generates the message honestly). Every affected recipient (and server) is then forced to pay extra costs to handle such spamming (just like if a mailbox is full of spam, we need to enlarge the mailbox to avoid not receiving regular mails).

Ideally, an OMR construction should be provably secure against such spamming attacks. In [34], a Regev05 [49] ciphertext included in each message is used to indicate pertinency[1] of that message: if the ciphertext decrypts to zero using a recipient's secret key, this recipient deems the corresponding message as pertinent. Thus, [34] proposes the *snake-eye conjecture*: any non-trivial Regev05 ciphertext cannot be decrypted to zero under two different secret keys (except with a trivial probability). Under this conjecture, the authors prove that such a spamming attack under the DoS model does not work, and refer to this property as DoS-resistance. However, the correctness of this conjecture has been an open problem. Fortunately, in this paper, we prove this conjecture to be true under a standard Learning With Error (LWE) lattice hardness assumption.

Another more consequential issue with the original [34] construction is its efficiency, mainly concerning the server runtime. To improve this, [36] proposes a new scheme with server runtime 15x faster than in [34]. While the efficiency is indeed greatly improved, [36] did not prove their construction to be DoS-resistant. In fact, we show concrete attacks that can spam *all* recipients (i.e., a maliciously crafted message can be detected as pertinent to all the honest recipients). Moreover, the attacks can be easily generalized, so such malicious messages cannot be easily fully rejected.

This raises the following question:

Can we construct an OMR scheme that is both provably DoS-resistant (under standard as-

[1]Or more generally, a PVW ciphertext [47].

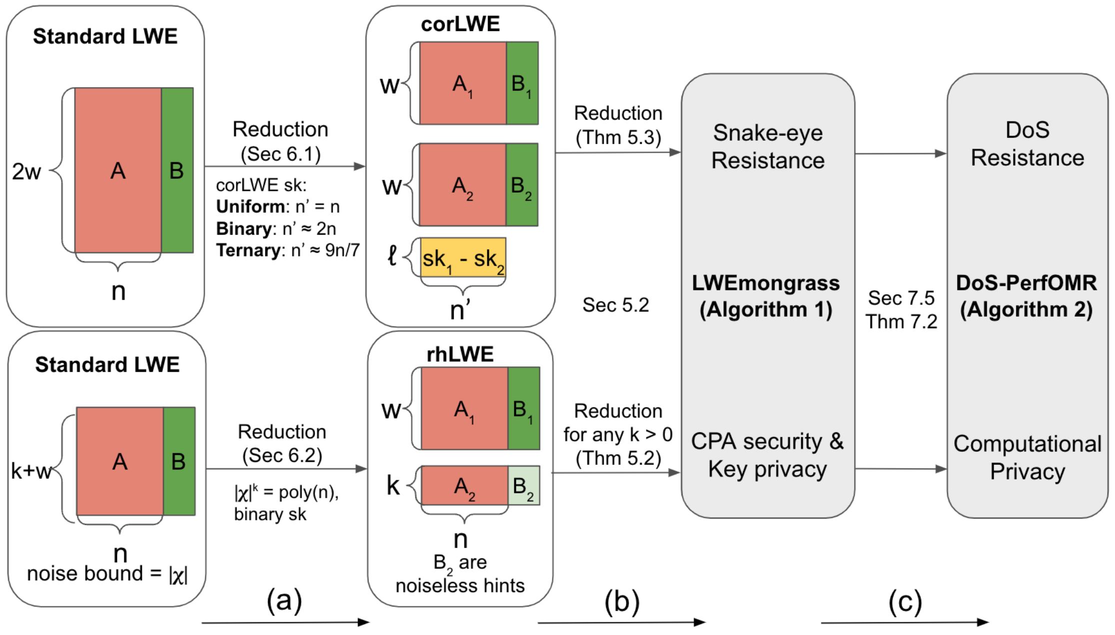

Figure 1: Overview of the security guarantee of our construction LWEmongrass (Corollary 6.7) and DoS-PerfOMR (Theorem 7.2). For standard LWE (Definition 3.1), $(A, B) \approx_c U$ (where U stands for uniform); for corLWE (Definition 4.2), $(A_i, B_i)_{i\in\{1,2\}} \approx_c U$, given $(\mathsf{sk}_1 - \mathsf{sk}_2)$; for rhLWE (Definition 5.1), a regular LWE sample $(A_1, B_1) \approx_c U$ given noiseless samples $(A_2, B_2 = A_2\mathsf{sk})$.

sumptions) and as efficient as the state of the art OMR construction without DoS-resistance (i.e., [36])?

In this paper, we indeed show such a scheme.

1.1 Our Contributions

Proving the Regev05/PVW snake-eye conjecture. We also resolve a related open problem posed by [34, Conj 8.4]: proving the Regev05 [49] snake-eye conjecture, assuming only standard LWE. This implies provable DoS-resistance for [34, 35]. Furthermore, we prove that PVW [47] is snake-eye resistant for *any* (polynomial) parameter under standard LWE, whereas [34] conjectured it (via a corollary) only for a small parameter regime.

Snake-eye resistance property of PKE. We generalize the snake-eye resistance property defined against Regev05 in [34] to a general property against all PKE schemes. When seeking more efficient lattice-based PKE schemes with snake-eye resistance, we show (by explicit attacks) that the natural candidates like short-key LWE encryption, or its Ring-LWE variant [38], or similar schemes like Crystals-Kyber [13], are all *not* snake-eye resistant. Consequentially, OMR schemes that employ these PKE schemes (e.g., [36]) are not DoS-resistant.

Improved snake-eye resistant lattice-based PKE scheme. We then construct a new PKE scheme LWEmongrass[2] that is provable snake-eye resistant and has better efficiency than PVW [47]. Its CPA security, key privacy, and snake-eye resistance are all proven under standard LWE for

[2]Lemongrass is folklore snake repellent.

the parameters we need to instantiate our scheme. Thus, it is also plausibly post-quantum secure.

LWE variants. We introduce two new LWE variants: *LWE with Correlation* (corLWE) and *LWE with Random Hints* (rhLWE), to prove the security and snake-eye resistance of the aforementioned schemes. The two assumptions are both LWE variants with side information: corLWE gives out two LWE samples under two secrets and the difference between the two secrets; rhLWE gives out an LWE sample with a random linear transformation of the underlying secret. We show that for the parameters needed to instantiate the schemes of interest, these two new variants are at least as hard as standard LWE (except for some small parameter loss). We believe those two new variants and the novel proof techniques we use for reductions are also of their own interest.

DoS-resistant OMR. Using the results above, we obtain for the first time practical OMR constructions that are DoS-resistant under standard assumptions. In particular, we first prove the DoS-resistance of the OMR construction in [34] under standard LWE hardness, and then obtain a more efficient DoS-resistant OMR construction by combining the aforementioned improved snake-eye-resistant PKE scheme $\mathsf{LWEmongrass}$ together with the techniques introduced in [36].

Implementation and evaluation. We implemented our OMR construction $\mathsf{DoS\text{-}PerfOMR}$ as a open-sourced C++ library [23] and measured its concrete performance in comparison to prior works. Compared to the OMR construction in [34], our construction is about 12x faster in terms of server (detector) runtime with 28x smaller public key size. Compared to the OMR construction in [36] (with *no* DoS-resistance), our construction has a comparable server runtime (about 1.2x slower), and the public key is about 2.2x larger.[3] We also experimentally show the effectiveness of the attacks above for the short-key lattice-based PKE schemes and against [36].

1.2 Technical Overview

Proving the Regev05/PVW snake-eye conjecture. To prove [34, Conj 8.4], we first introduce a new variant of LWE, called LWE with correlation (corLWE), which gives out two regular LWE samples under two different secrets *and* the difference between the two secrets. We later show that corLWE is at least as hard as standard LWE. For now, we directly use corLWE to prove the snake-eye conjecture as follows: an adversary that breaks the conjecture can generate a ciphertext $\mathsf{ct} = (\vec{a}, b)$ that decrypts to 0 under two independent and valid secret keys, sk_1 and sk_2 (i.e., for $i \in [1,2]$, $b_i - \langle \vec{a}, \mathsf{sk}_i \rangle \approx 0$). Based on the linearity of decryption, the adversarial ciphertext decrypts to 0 with noticeable probability when using as key the difference of the two secrets (i.e., $\langle \vec{a}, \mathsf{sk}_1 - \mathsf{sk}_2 \rangle \approx 0$). However, this is not the case if instead of two valid public keys (i.e., valid LWE samples), the adversary is given uniform samples in dependent of the secret difference (except for a trivial ciphertext with $\vec{a} = \vec{0}$). Thus, this adversary breaks the corLWE assumption.

Snake-eye resistance property of short-key LWE PKE schemes. We then extend the original snake-eye conjecture of [34] to formally define a general snake-eye resistance property for PKE: any (non-trivial) ciphertext cannot be decrypted to the same value under two honestly generated secret keys. We analyze this property on existing lattice-based PKE schemes and show that the schemes that are based on short secrets (e.g., [13, 38]) are *not* snake-eye resistant. When the secret keys are short, we can construct wildcard ciphertexts that decrypt to 0 for most honestly generated secret keys. For example, consider the ciphertext $(\vec{a} = \vec{1}, b = 2n/3) \in \mathbb{Z}_q^{n+1}$ (where n is

[3]The measured performance of our scheme is actually better than all prior OMR benchmarks, DoS-resistant or otherwise, when we utilize Intel AVX-512 CPU acceleration. $\mathsf{PerfOMR}$ [36] closes the gap when we likewise accelerate it, but is not DoS-resistant. See Section 8.

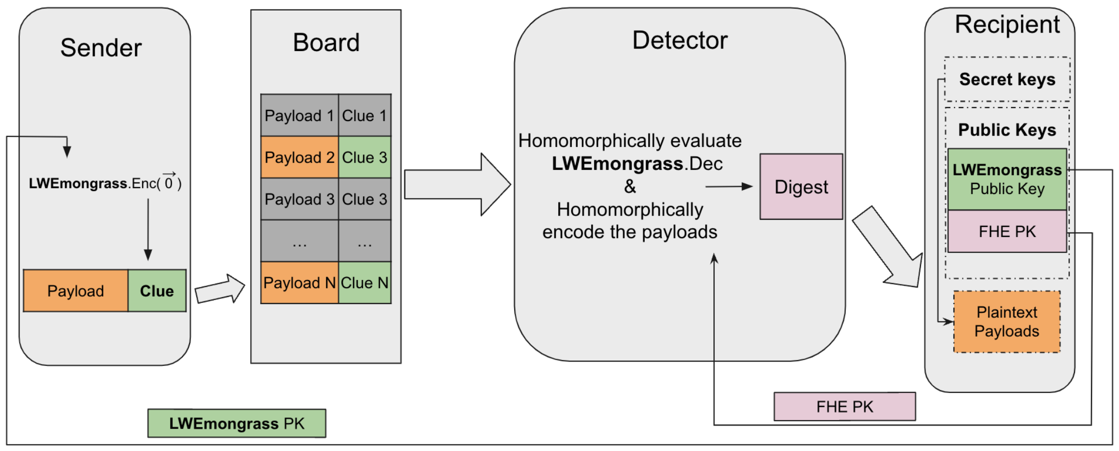

Figure 2: Overview of building our DoS-resistant OMR scheme DoS-PerfOMR from our snake-eye resistance PKE scheme LWEmongrass. FHE stands for fully homomorphic encryption, for which we use the BFV scheme (Section 3.6). DoS-resistance and privacy guarantees of DoS-PerfOMR are shown in Fig. 1 step (c).

the length of the secret key and $q \gg n$), a uniform ternary secret (i.e., all the elements of the secret are in $\{-1, 0, 1\}$) decrypts c to 0 with high probability, since $b - \langle \vec{a}, \mathsf{sk} \rangle \approx 0$ for any ternary sk. This attack easily generalizes to other ciphertext forms (i.e., $\vec{a} \neq \vec{1}$) and thus is hard to prevent.

A more efficient snake-eye resistant lattice-based PKE scheme. To achieve provable snake-eye resistance while maintaining the efficiency of short-key LWE PKE schemes, we introduce our new PKE scheme, LWEmongrass, that uses a hybrid of uniform and short secrets. The uniform component of the secret guarantees the snake-eye resistance under corLWE (also with a hybrid secret). To accommodate the uniform secret elements, the encryption procedure of LWEmongrass needs to generate some noiseless LWE samples. Otherwise, the decryption error becomes too large (as the error terms are multiplied by the secret key elements during decryption). Therefore, to prove the security of LWEmongrass, we introduce another LWE variant LWE with random hints (rhLWE), which gives a regular LWE sample together with some random linear combinations of the secret (i.e., the noiseless terms). Under rhLWE (with short secrets), we prove that LWEmongrass is CPA-secure and key-private. Fig. 1 step (b) shows how LWEmongrass achieves its desired properties.

Reducing standard LWE to corLWE. For corLWE with uniform secrets, the reduction incurs almost no security loss: given a regular LWE sample with $2w$ components, we separate them into two regular LWE samples each with w components, and randomly generate a uniform secret to decouple the second sample by masking the underlying uniform secret. With these two samples and the new uniform secret, we can construct a valid corLWE challenge. This thus proves [34, Conj 8.4] under standard LWE, combining with the result above.

However, for LWEmongrass, recall that we need corLWE to hold with short secrets. First, observe that the above strategy fails for uniform binary secrets, since we cannot use a binary secret to mask another binary secret(the resulting secret is no longer binary). Instead, observe that if two independently sampled secrets have the same bit (0 or 1) at location i, there is a one-bit entropy even when the difference between the two secrets is given out (as the difference at location i is always 0). Each such location then allows us to embed a uniform binary LWE secret element.

Hence, to embed an LWE secret of size n, we create a corLWE instance with secrets of size $n' \approx 2n$ (also see the upper segment of Fig. 1 step (a)).

Reduction for ternary secret is a bit more involved: using the same argument above for ternary secret results in $n' \approx 3n$ parameter loss, which is not tight. We observe that for ternary secrets, the difference (of location i) being 0 preserves a $\log_2(3)$-bit entropy, and the difference being 1 or -1 also preserves one-bit entropy. To leverage this observation, we reduce standard LWE with a hybrid of ternary and binary secrets to corLWE with ternary secrets and obtain $n' \approx 9n/7$.

Reducing standard LWE to rhLWE. To reduce standard LWE (with binary secrets) to rhLWE with binary secrets, we use a proof technique inspired by [27]: given an LWE sample with $w + k$ components, we guess the error of the last k components (again using the notation in Fig. 1). If we guess it correctly, the error cancels out and we get a valid rhLWE sample. Otherwise, our construction gives a random sample. As long as we guess correctly with $1/\mathsf{poly}(n)$ probability, we can break standard LWE using an adversary that breaks rhLWE (also see the lower segment of Fig. 1 step (a)).

With these two reductions, we show desired properties of $\mathsf{LWEmongrass}$ are proven under standard LWE assumption (Corollary 6.7).

Applying LWEmongrass to construct OMR. To construct an efficient OMR scheme (overview illustrated in Fig. 2 and Fig. 1 step (c)), we follow the framework introduced in [34]: a $\mathsf{LWEmongrass}$ ciphertext is included in a message to (implicitly) indicate its recipient (ciphertexts decrypting to 0 indicate pertinency). The untrusted server uses FHE to homomorphically decrypt the ciphertext and result (under FHE) contains whether the recipient performing the retrieval is the intended recipient. Since $\mathsf{LWEmongrass}$ has a similar setup as regular short-key LWE PKE schemes, we can leverage the optimizations (particularly designed for short-key LWE PKE schemes) introduced in [36] for the homomorphic retrieval circuit. Combining all these, we obtain our OMR scheme $\mathsf{DoS\text{-}PerfOMR}$ that is about 12x faster than the constructions in [34, 35] (proven to be DoS-resistant by our result).

1.3 Organization

The rest of the paper is organized as follows. In Section 3, we introduce the main preliminary knowledge needed for our paper. In Section 4, we formally define the snake-eye resistance property and prove that Regev05/PVW is snake-eye resistant. In Section 5, we show that some other efficient LWE-based PKE schemes are not snake-eye resistant, and construct a new PKE scheme that is snake-eye resistant and more efficient than PVW. In Section 6, we show the two LWE variants proposed are equivalent to LWE under the parameters of interest. In Section 7, we recall the definition of DoS-resistant OMR and show how to apply our results about the snake-eye-resistant PKE scheme to achieve DoS-resistant OMR. In Section 8, we implement our DoS-resistant OMR and compare it with prior constructions; we also demonstrate our attack on prior constructions. In Section 9, we discuss the relation between our snake-eye resistance property and the robustness property of PKE. In Section 10, we discuss some extensions on the snake-eye resistance property. In Section 11, we discuss some open questions.

2 Related Works

Oblivious Message Retrieval. OMR [34] proposes a message retrieval primitive with full recipient privacy. Later GOMR [35] extends it to the group setting. Both works rely on a hybrid use of the PVW encryption scheme and BFV-leveled homomorphic encryption scheme. The DoS-resistance of both works is conjectured without proof under standard assumptions.

PerfOMR [36] replaces the PVW encryption scheme with a tailored RLWE encryption scheme and modifies the underlying BFV-based retrieval circuit. It achieves a better efficiency compared to [34, 35] in terms of detector runtime, clue key size, and clue size. However, PerfOMR is not DoS-resistant. We show an attack to it in Section 5.1.

One recent concurrent and independent work [11] proposes a general DoS-resistant OMR construction using FHE and commitment schemes. However, unfortunately, their construction relies on heavy machinery: their clue includes encryption of an FHE public key and a commitment to that public key. Thus, the clue needs to be at least megabytes, compared to the schemes above mainly with clue < 1KB. The detector needs to perform a decryption of the encrypted public key, a commitment on the decrypted public key, and a comparison between the new commitment and the original one in the clue, all homomorphically. In contrast, other OMR works [34, 35, 36] only require a homomorphic decryption of ℓ bits for some small constant ℓ (e.g., ≤ 6). Based on our conservative estimation, realizing their scheme requires $\gg 200$ levels of multiplication, while a practical OMR scheme like [34, 35, 36] requires only about 20 levels; the number of homomorphic operations is even worse. Therefore, the runtime and the clue size of concretely realizing this work can be orders of magnitude worse than [34, 35, 36] (either with or without using bootstrapping). Therefore, while [11] provides an interesting theoretical possibility, due to its inherent heaviness, it is far from being practical.[4]

Another concurrent and independent work [31] shows an OMR-like construction that is DoS-resistant (among other features), based on two non-colluding servers OMR construction. However, it achieves a weaker security notion: for privacy, it assumes that each individual server is *semi-honest even without collusion* (i.e., not only they do not collude with each other, but they also cannot collude with a recipient or simply pretend to be a recipient). Violations of these assumptions would lead to undetectable privacy failure. Conversely, our work together with all the prior works [34, 35, 36, 11, 39, 30] and the private signaling papers below [39, 30] ensure privacy against malicious servers (and assume semi-honest servers just for correctness, whose violation may be detected or prevented by other mechanisms such as consistency checking of SNARK proofs). Also, their construction is not compact (i.e., communication linear in the board size) and needs an additional round of PIR to fetch the payloads. They introduce another additional interesting notion called "deletion": that the servers delete the messages they store for every certain period. This operation can improve efficiency as the size of the bulletin board. However, this operation allows the servers to know which messages are retrieved during this period, which may lead to extra information leakage.

Fuzzy Message Detection. FMD [7, 48] mainly focuses on decoy based security. While this primitive has highly efficient constructions, we consider the security notion relatively weak as analyzed in [51]. Therefore, we do not compare these constructions directly.

Private Signaling. Like OMR, PS [39, 30] provides full security, but it only supports detection

[4]Motivated by another application, their construction provides a stronger DoS guarantee: it holds even if the attacker knows recipients' secret keys.

instead of retrieval (i.e., letting the recipients know the indices of pertinent messages instead of the payloads). Furthermore, PS as introduced in [39] does not require any form of key-unlinkability; this is in contrast to the OMR security guarantees. For instance, in the OMR setting detection-key-unlinkability as defined in [35] states that retrievals cannot be linked to a particular recipient. Later works [30] enhanced the definition of [39] to offer detection-key-unlinkability, but still not full-key-unlinkability as requires in [34] (i.e., OMR allows a recipient to hold multiple clue keys that cannot be linked). Lastly, PS does not require the property of DoS resistance, which is the main topic of this work. See also [34, Sec 2.3] for a detailed comparison between OMR and PS.

In terms of constructions, prior works on private signaling have constructions using a Trusted Execution Environment (TEE), which is a strong environment assumption since a lot of work shows that the existing TEEs have side-channels that can leak secrets easily [52]. Therefore, while the construction in [30] provides a construction with great scalability (the runtime growth is only poly-logarithmic in the number of messages), we do not directly compare to them as we are assuming a standard environment. [39] provides a solution assuming two communicating but non-colluding servers, which is also a strong environment assumption. Moreover, this construction also scales linearly the number of messages and is concretely slower than [34] and our constructions. Therefore, we do not compare with this construction directly either.

Other private retrievals primitives. There are other related problems like *(batch) Private Information Retrieval (PIR)* [20] and its variant *(batch) Keyword PIR* [19] and Private Stream Search (PSS) [45, 22, 9, 26]. They share some similarities with OMR but are also very different from it as discussed in [34, 35, 36].

LWE variants. The two LWE variants we introduced are with side information, while sharing similarities with variants in prior works [44, 5, 17, 3, 33, 18, 32, 28, 27, 16], our variants differ in terms of definition and proof techniques. We discuss the relationship between those variants and our variants in Section 6.3 in more detail, after we formally define the two variants we introduce.

Robust encryption. Robust encryption introduced in [2] (motivated by private auction [50] and consistency in searcheable encryption [1]) requires that a ciphertext can only be considered valid for one honestly generated secret key. By definition, it is stronger than the snake-eye resistance property we define in Definition 4.1, so not required for our OMR application. Furthermore, prior works use heavier machinery which is inefficient in OMR context. We show that a snake-eye-resistant PKE (like ours) implies a robust PKE. via an efficient black-box transformation. Thus, our construction can also serve as new ways to build new robust PKE schemes with better efficiency in some parameter regimes. See Section 9 for a more detailed discussion.

3 Preliminary

3.1 Notation

Let $\mathcal{D}$ denote an arbitrary distribution, $\mathcal{B}$ denote the binary distribution, $\mathcal{T}$ denote the ternary distribution (i.e., the uniform distribution over $\{-1,0,1\}$), $\mathcal{U}$ denote the uniform distribution. Let $x \xleftarrow{\$} P$ denotes a uniformly random sample from space P, $x \leftarrow \mathcal{D}$ denotes a uniform sample from distribution $\mathcal{D}$. Let $\mathcal{D}_\beta$ denote a distribution with norm bound β, s.t. $\Pr_{x\leftarrow\mathcal{D}_\beta}[|x| > \beta] \leq \mathsf{negl}(\lambda)$ for security parameter λ. Let $|\chi|$ denote the norm bound of a symmetric distribution χ such that $\Pr[x = i] = \Pr[x = -i]$ for all $0 \leq i \leq |\chi|$. Finally, $\mathcal{D}^\ell$ represents the product distribution $\mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}$ (e.g. if $\vec{x} \leftarrow \mathcal{D} \in \mathbb{Z}_q^n$, then $X \leftarrow \mathcal{D}^\ell \in \mathbb{Z}_q^{n\times\ell}$).

For $x \in \mathbb{Z}_q$, let $|x|$ denote the norm of x when lifted to $[-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$; in other words, $|x| = |z|$ where $z \in [-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$ satisfying $z + x \mod q = 0$. Let $\vec{z}$ for $z \in \mathbb{Z}$ denote a vector with all elements being z, i.e. $(z, z, \ldots, z)$ (e.g., $\vec{1} := (1, 1, \ldots, 1)$). Let 0^k be a vector of $k \in \mathbb{Z}^+$ zeros. Similarly, $0^{k \times k'}$ is a matrix of $k \times k'$ zeros for $k, k' \in \mathbb{Z}^+$. For a vector $\vec{x}$, $\vec{x}[a, b]$ denotes a vector containing the a-th element up to the b-th element in $\vec{x}$. For a matrix $A \in \mathbb{Z}_t^{m \times n}$, $A^\intercal \in \mathbb{Z}_t^{n \times m}$ denotes its transpose.

3.2 Hard Problem

Definition 3.1 (Decisional Learning With Error Assumption)**.** Let $n, \ell, w, q, \mathcal{D}, \chi$ be parameters dependent on λ. The (decisional) learning with error (LWE) assumption $\mathsf{LWE}_{n,\ell,w,q,\mathcal{D},\chi}$ states the following: any PPT adversary cannot distinguish $(A, AS + E)$ and (A, B) (except with negligible advantage), where $A \xleftarrow{\$} \mathbb{Z}_q^{w \times n}$, $S \leftarrow \mathcal{D}^{n \times \ell}$, $E \leftarrow \chi^{w \times \ell}$ and $B \xleftarrow{\$} \mathbb{Z}_q^{w \times \ell}$.

3.3 Public Key Encryption

Definition 3.2 (Public Key Encryption (PKE))**.** A PKE scheme consists of four PPT algorithms ($\mathsf{PKE.GenParam}, \mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec}$) defined as follows:

- $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$: takes a security parameter λ, an auxiliary parameter aux with $|\mathsf{aux}| = \mathsf{poly}(\lambda)$; outputs a public parameter pp.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp})$: takes the public parameter pp; outputs a key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{pp}, \mathsf{pk}, m)$: takes the public parameter pp and the public key pk, generates the ciphertext c for the message m; outputs ct.
- $m \leftarrow \mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$: takes the public parameter pp, decrypts the ciphertext ct into a plaintext message m based on the secret key sk; outputs the message m.

The scheme must satisfy the following properties:

- (Correctness) For any valid aux, let $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$, let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}($ $\mathsf{pp})$ and for any message m, let $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{pp}, \mathsf{pk}, m)$ and $m' \leftarrow \mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$. It holds that if $\mathsf{pp} \neq \perp$, $\Pr[m = m'] \geq 1 - \mathsf{negl}(\lambda)$.
- (IND-CPA Security) For any PPT algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$: let $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp})$; $\mathcal{A}_1$ chooses two messages and remembers state: $(m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})$; sample $b \xleftarrow{\$} \{0,1\}$ and $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{pp}, \mathsf{pk}, m_b)$; let $b' \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct})$; it holds that: $\Pr[b = b'] \leq 1/2 + \mathsf{negl}(\lambda)$.
- (Key privacy) For any PPT algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$: let $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$, and $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp}), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp})$; $\mathcal{A}_1$ chooses a message and remembers its state: $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk}_0, \mathsf{pk}_1)$; then sample $b \xleftarrow{\$} \{0,1\}$ and let $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{pp}, \mathsf{pk}_b, m)$; let $b' \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct})$; it holds that: $\Pr[b = b'] \leq 1/2 + \mathsf{negl}(\lambda)$.

3.4 PVW PKE scheme

Definition 3.3. We describe the PVW scheme introduced in [47] (adapted according to [34]).

- $\mathsf{pp} = (n,\ell,w,q,p,\chi,r) \leftarrow \mathsf{PVW.GenParam}(1^\lambda, \mathsf{aux} = (\ell,q,p,\chi))$: Choose a secret key dimension n and public key dimension $w = \mathsf{poly}(\lambda,n,\ell,q)$ as in [47], where q is the ciphertext modulus, p is the plaintext modulus, and ℓ is the number of $\mathbb{Z}_p$ elements in plaintexts. Finally, set r to be minimum $r > 0$ such that $\Pr_{e_i \xleftarrow{\$} \chi}[\sum_{i\in[w]} |e_i| > r] \le \mathsf{negl}(\lambda)$, where χ is the input noise distribution; if $r > \frac{q}{2p}$, output $\mathsf{pp} = \perp$[5].

- $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{PVW.KeyGen}(\mathsf{pp})$: If $\mathsf{pp} = \perp$, output $(\mathsf{pk},\mathsf{sk}) = (\perp,\perp)$. Otherwise, sample a random secret key $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_q^{n\times\ell}$, a random $A \xleftarrow{\$} \mathbb{Z}_q^{w\times n}$ and an error matrix $E \leftarrow \chi_\sigma^{w\times\ell}$, and compute $\mathsf{pk} = (A, P = A\ \mathsf{sk} + E)$.

- $\mathsf{ct} = (\vec{a},\vec{b}) \leftarrow \mathsf{PVW.Enc}(\mathsf{pp},\mathsf{pk},\vec{m})$: If $\mathsf{pp} = \perp$, output $\mathsf{ct} = \perp$. Otherwise, to encrypt a vector $\vec{m} \in \mathbb{Z}_p^\ell$, define a vector $\vec{t} = \lfloor \frac{q}{p} \rfloor \cdot \vec{m} \in \mathbb{Z}_q^\ell$. Sample a random row vector $\vec{x} \xleftarrow{\$} \{0,1\}^{1\times w}$. Output $(\vec{a},\vec{b}) = (\vec{x}A, \vec{x}P + \vec{t}) \in \mathbb{Z}_q^{1\times n} \times \mathbb{Z}_q^{1\times\ell}$ as the ciphertext.

- $\vec{m} \leftarrow \mathsf{PVW.Dec}(\mathsf{pp},\mathsf{sk},\mathsf{ct} = (\vec{a},\vec{b}))$: If $\mathsf{pp} = \perp$, output $\vec{m} = \perp$. If $\vec{a} = 0^n$, output $\vec{m} = \perp$. Otherwise, let $\vec{d} = (\vec{b} - \vec{a}\mathsf{sk})^\intercal \in \mathbb{Z}_q^\ell$, output $\vec{m} \in \mathbb{Z}_p^\ell$, where $\vec{m}[i] = v$ if there exists $v \in \mathbb{Z}_p$ such that: $\left|\left\lfloor \vec{d}[i] \cdot \frac{p}{q} \right\rfloor - v\right| \le r$; otherwise, $\vec{m}[i] = \perp$.

This PVW PKE scheme is correct, CPA-secure, and key-private as shown in [47, 34] under $\mathsf{LWE}_{n,\ell,w,q,\mathcal{U}^n,\chi_\sigma}$. Fixing $\ell = 1$ gives the Regev05 [49] scheme.

Remark 3.4. The two major differences between this variant and the original PVW [47] are: (1) r in [47] is strictly $q/2p$ while in this variant, it can be smaller; (2) when $\vec{a} = 0^n$, this variant rejects to decrypt the ciphertext. Both changes are proposed in [34]: change (1) is to reduce the range to check for efficiency (smaller r reduces ℓ in the application of [34]); change (2) avoids a trivial ciphertext that decrypts to the same plaintext for any secret key. It is straightforward that (1) does not affect correctness, CPA-security, or key-privacy. For (2), since an honestly generated ciphertext under an honestly generated public key gets rejected with probability $q^{-n} = \mathsf{negl}(n)$, it has negligible effect over correctness, CPA-security, or key-privacy.

3.5 shortLWE PKE scheme

Now we recall the shortLWE PKE scheme introduced in [38] (adapted according to [36]). This scheme follows the blueprint of PVW except that the secrets are sampled from a distribution $\mathcal{D}_\beta$ with some (small) norm bound $\beta \ll q$. Also, instead of sampling a subset-sum of the public key inside Enc, the encryption procedure samples an LWE secret to generate new LWE samples. We show these differences (with respect to Definition 3.3) in blue color. Note that in [38, 36], the scheme is constructed based on the Ring-LWE assumption. For the purpose of this paper, we adapt the scheme to regular LWE.

Definition 3.5. shortLWE is defined as follows.

[5]In [34], r is chosen based on the condition $\Pr_{e_i \xleftarrow{\$} \chi_\sigma, x_i \xleftarrow{\$} \{0,1\}}[\sum_{i\in[w]} x_i \cdot |e_i| > r] \le \mathsf{negl}(\lambda)$. We simplify this which suffices for our applications.

- $\mathsf{pp} = (n,\ell,q,p,\chi,r) \leftarrow \mathsf{shortLWE.GenParam}(1^\lambda, \mathsf{aux} = (\ell, \mathsf{q}, \mathsf{p}, \mathcal{D}_\beta, \chi))$: Choose a secret key dimension n so that the $\mathsf{LWE}_{n,\ell,n+\ell,q,\mathcal{D}_\beta^n,\chi}$ assumption holds, where q is the ciphertext modulus, p is the plaintext modulus, ℓ is the number of $\mathbb{Z}_p$ elements in plaintexts, χ is the noise distribution, and $\mathcal{D}_\beta^n$ is the secret key distribution. Finally, set r to be minimum $r > 0$ such that $\Pr_{x_i \xleftarrow{\$} \chi_\sigma}[\sum_{i\in[2n+1]} |x_i| \cdot \beta > r] \leq \mathsf{negl}(\lambda)$ where β is the bound of the distribution $\mathcal{D}_\beta$; if $r > \frac{q}{2p}$, output $\mathsf{pp} = \perp$. [6]

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{shortLWE.KeyGen}(\mathsf{pp})$: If $\mathsf{pp} = \perp$, output $(\mathsf{pk}, \mathsf{sk}) = (\perp, \perp)$. Otherwise, sample a random secret key $\mathsf{sk} \leftarrow \mathcal{D}_\beta^{n\times\ell}$, a random $A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ and an error matrix $E \leftarrow \chi_\sigma^{n\times\ell}$. Compute $\mathsf{pk} = (A, P = A\ \mathsf{sk} + E)$.

- $\mathsf{ct} = (\vec{a}, \vec{b}) \leftarrow \mathsf{shortLWE.Enc}(\mathsf{pp}, \mathsf{pk}, \tilde{\mathsf{m}})$: If $\mathsf{pp} = \perp$, output $\mathsf{ct} = \perp$. Otherwise, to encrypt a vector $\vec{m} \in \mathbb{Z}_p^\ell$, define a vector $\vec{t} = \lfloor \frac{q}{p} \rfloor \cdot \vec{m} \in \mathbb{Z}_q^\ell$. Sample $\vec{x} \xleftarrow{\$} \mathcal{D}_\beta^{1\times n}$, $\vec{e}_1 \leftarrow \chi_\sigma^{1\times n}$, $\vec{e}_2 \leftarrow \chi_\sigma^{1\times\ell}$. Output $(\vec{a}, \vec{b}) = (\vec{x}A + \vec{e}_1, \vec{x}P + \vec{e}_2 + \vec{t}^\intercal) \in \mathbb{Z}_q^{1\times n} \times \mathbb{Z}_q^{1\times\ell}$ as the ciphertext.

- $\vec{m} \leftarrow \mathsf{shortLWE.Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct} = (\vec{a}, \vec{b}))$: If $\mathsf{pp} = \perp$, output $\vec{m} = \perp$. If $\vec{a} = 0^n$, output $\vec{m} = \perp$. Otherwise, let $\vec{d} = (\vec{b} - \vec{a}\mathsf{sk})^\intercal \in \mathbb{Z}_q^\ell$, output $\vec{m} \in \mathbb{Z}_p^\ell$, where for all $i \in [\ell]$, $\vec{m}[i] = v$ if there exists $v \in \mathbb{Z}_p$ such that: $\left| \left\lfloor \vec{d}[i] \cdot \frac{p}{q} \right\rceil - v \right| \leq r$; otherwise, $\vec{m}[i] = \perp$.

The $\mathsf{shortLWE}$ PKE scheme is correct, CPA-secure, and key-private under $\mathsf{LWE}_{n,\ell,n+\ell,q,\mathcal{D}_\beta^n,\chi}$.

3.6 BFV Homomorphic Encryption Scheme

The BFV [15, 25] homomorphic encryption scheme is a PKE scheme (Definition 3.2) that also allows homomorphic evaluation. That is, for two BFV ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ encrypting $m_1, m_2 \in \mathbb{Z}_t$ for some t respectively, let $\mathsf{ct}' \leftarrow \mathsf{ct}_1 \circ \mathsf{ct}_2$, it satisfies that $\mathsf{Dec}(\mathsf{ct}') = m_1 \circ m_2$ where operation $\circ \in \{\times, +\}$ (homomorphic evaluation on op is done using the BFV public key $\mathsf{BFV.pk}$). We assume that BFV is unconditionally correct and secure (under Ring-LWE assumption) in this paper. We basically use BFV as a black box for our OMR construction and thus omit further details.

3.7 The Snake-eye Conjecture

We state the snake-eye conjecture introduced in [34, Conj 8.4]. This conjecture motivates our general snake-eye definition for PKE schemes and we later prove a slightly different version of this conjecture (Remark 4.4). The conjecture essentially says that for Regev05 PKE scheme [49] (which can also be seen as the PVW scheme [47] with $\ell = 1$), any non-trivial ciphertext cannot be decrypted to 0 for two different keys except with trivial probability. As discussed in [34], this property is not implied by standard security notions like CPA or CCA security or key privacy.

Conjecture 3.6 (Regev05 is snake-eye resistant)**.** For any PPT algorithm $\mathcal{A}$, for Regev05 encryption with modulus q and remaining parameters for which semantic security holds, for any $1 \leq r < q/4$, key pairs $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $(\mathsf{sk}', \mathsf{pk}') \leftarrow \mathsf{KeyGen}(1^\lambda)$ and ciphertext $(\vec{a}, b) \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{pk}', r)$,

[6] In [36], r is chosen by $\Pr_{e_i,e_j \xleftarrow{\$} \chi_\sigma, x_i \xleftarrow{\$} \mathcal{D}_\beta}[\sum_{i\in[2n]} x_i \cdot |e_i| + \sum_{j\in[\ell]} |e_j| > r] \leq \mathsf{negl}(\lambda)$. The simplified condition suffices for our results.

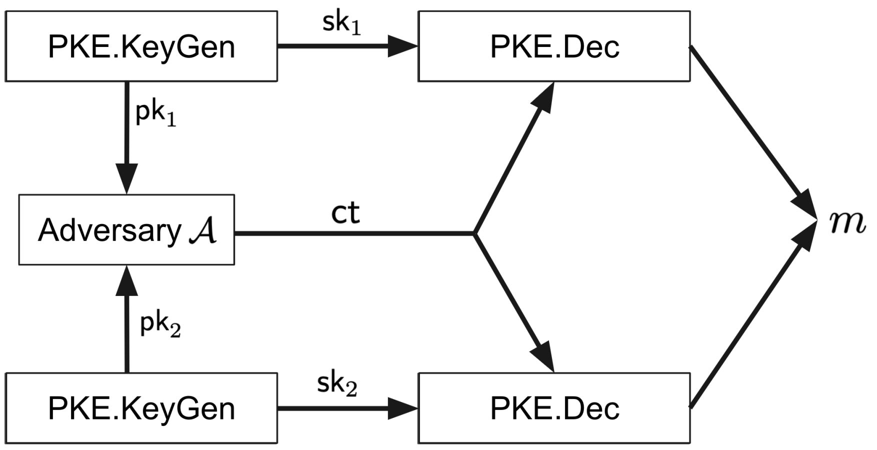

Figure 3: Overview of definition of δ-snake-eye resistance, where δ is the success probability that the adversary wins the game in the figure. m is any plaintext in the plaintext space.

it holds that

$$\Pr\left[\begin{array}{c} |\vec{a}\ \mathsf{sk} + b| \leq r \\ \wedge\ |\vec{a}\ \mathsf{sk} + b| \leq r \end{array} \wedge\ \vec{a} \neq \vec{0}\right] \ \leq\ (2r+1)/q + \mathsf{negl}(\lambda)\ .$$

Then, [34, Lemma 8.5] shows that PVW [34] is snake-eye resistant given that Conjecture 3.6 holds, for $((2r+1)/q)^{-\ell} = \mathsf{poly}(\lambda)$, which means $\ell = O(\log(\lambda))$ (as $((2r+1)/q) \geq 2$ is needed for correctness, and it can be even smaller if $(2r+1)/q$ is $\omega(1)$), formally stated as follows.

Lemma 3.7 ([34] PVW is snake-eye resistant)**.** *Under Conjecture 3.6, for any PPT adversary* $\mathcal{A}$*, for* PVW *encryption with ciphertext modulus* q *and plaintext space* $\mathbb{Z}_2^\ell$*, and* r *such that* $((2r+1)/q)^{-\ell} = \mathsf{poly}(\lambda)$ *and remaining parameters for which semantic security holds, for any* $1 \leq r < q/4$*, for key pairs* $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{PVW.KeyGen}(\mathsf{pp})$ *and* $(\mathsf{sk}',\mathsf{pk}') \leftarrow \mathsf{PVW.KeyGen}(\mathsf{pp})$*, for ciphertext* $c = (\vec{a},\vec{b}) \leftarrow \mathcal{A}(\mathsf{pk},\mathsf{pk}',r)$*, letting* $\vec{m} \leftarrow \vec{b}^{\mathsf{T}} - \mathsf{sk}^{\mathsf{T}}\vec{a}^{\mathsf{T}}$ *and* $\vec{m}' \leftarrow \vec{b}^{\mathsf{T}} - \mathsf{sk}'^{\mathsf{T}}\vec{a}^{\mathsf{T}}$*, it holds that:*

$$\Pr\Big[\big(\forall i \in [\ell] : |m[i]| \leq r \wedge |m'[i]| \leq r\,\big) \wedge\ \vec{a} \neq \vec{0}\Big]\ \leq\ \big((2r+1)/q\big)^{\ell} + \mathsf{negl}(\lambda)\ .$$

4 Snake-eye Resistant Public Key Encryption

We start by answering the open question raised in [34]: proving that Regev05 is snake-eye resistant (Conjecture 3.6 above) under a standard LWE hardness assumption, and likewise for PVW (Definition 3.3). This implies the DoS-resistance of [34]. Then, we build our own more efficient PKE scheme using insights from the proof, which is then used to construct the more efficient DoS-resistant OMR. To prove Conjecture 3.6, we prove a more general version: PVW PKE scheme (Definition 3.3) is snake-eye resistant (which implies that Regev05 is snake-eye resistant). To accomplish the proof, we first generalize the definition to discuss *snake-eye resistance* of PKE schemes in general.

4.1 PKE Snake-eye Resistance

At a high-level, we define δ-snake-eye resistance to be the property that any ciphertext cannot be decrypted into the same plaintext under two honestly generated keys except with δ probability (visualized in Fig. 3). The formal version is as follows.

Definition 4.1 (δ-snake-eye resistance of PKE)**.** A PKE scheme is δ-snake-eye-resistant if the following holds: let $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp}), (\mathsf{pk}_2,\mathsf{sk}_2) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp})$; for any PPT adversary $\mathcal{A}$, let $\mathsf{ct} \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{pk}_2)$:

$$\Pr\left[\mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_1, \mathsf{ct}) = \mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_2, \mathsf{ct}) \neq \perp\right] \leq \delta.$$

Further generalizing snake-eye resistance. One may wonder whether this property can be further generalized. We discuss it in more detail in Section 10.

4.2 Snake-eye resistance of PVW

We proceed to prove that Regev05 is $(\frac{4r+1}{q})$-snake-eye-resistant, and more generally, PVW is $(\frac{4r+1}{q})^\ell$-snake-eye-resistant. To prove these, we employ a new LWE assumption variant called *LWE-with-correlation (*corLWE*)*. Later in Section 6.1, we show that this LWE-variant is equivalent to standard LWE (Definition 3.1) except for some small parameter loss. Looking ahead, all the properties we prove in this paper are under the standard LWE assumption. Meanwhile, we still formally define this variant due to the following reasons: (1) this new assumption is of its own interest and might have other applications; (2) we only prove that this assumption is equivalent to standard LWE under certain parameters, and for other parameter choices, it may worth further exploration; (3) it is more straightforward to prove the snake-eye resistance property directly under this new assumption (and later argue its equivalence to regular LWE under the parameters of interest).

At a high-level, corLWE says that given two LWE samples under two different secrets, together with the *difference* of the two secrets, the LWE samples are still indistinguishable from uniformly random samples. It is formally defined as follows.

Definition 4.2 (LWE with Correlation)**.** Let $n, \ell, w, q, \mathcal{D}, \chi$ be parameters dependent on λ.The decisional LWE with correlation assumption $\mathsf{corLWE}_{n,\ell,w,q,\mathcal{D},\chi}$ states the following: any PPT adversary cannot distinguish $(A_1, A_1\ \mathsf{sk}_1 + E_1, A_2,A_2\ \mathsf{sk}_2 + E_2, \mathsf{sk}_1 - \mathsf{sk}_2)$ from $(A_1, B_1, A_2, B_2, \mathsf{sk}_1 - \mathsf{sk}_2)$ (except with negligible advantage), where $A_1, A_2 \overset{\$}{\leftarrow} \mathbb{Z}_q^{w\times n}$, $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow \mathcal{D}^\ell, E_1, E_2 \leftarrow \chi^{w\times\ell}$, and $B_1, B_2 \overset{\$}{\leftarrow} \mathbb{Z}_q^{w\times\ell}$.

We then prove that PVW is snake-eye resistant under corLWE with secret key distribution $\mathcal{D} = \mathcal{U}^n$ (i.e., secrets are drawn uniformly at random from $\mathbb{Z}_q^n$)[7], For simplicity, we focus on the $\ell = 1$ case: by definition, the adversary breaking PVW snake-eye resistance gives out a ciphertext $(\vec{a}, b)$ that satisfies $|\vec{a}\ \mathsf{sk}_i - b| \leq r + d$ for $i \in \{1,2\}$, and d being (the encoding of) some plaintext. If we know the difference $\mathsf{sk}_1 - \mathsf{sk}_2$ (which is given in the corLWE challenge) and the maliciously crafted ciphertext, it is simple to test whether $|\vec{a}\ (\mathsf{sk}_1 - \mathsf{sk}_2)| \leq 2r$. If the corLWE challenge is random, this happens with probability $\frac{4r+1}{q}$. Therefore, if $\mathcal{A}$ has noticeable advantage over $\frac{4r+1}{q}$ for a normal corLWE challenge, it breaks the corLWE assumption. This naturally extends to the $\ell > 1$ case. We formalize the lemma and the proof as follows.

[7]Interestingly, for uniform secrets, there is almost no security loss when reducing regular LWE to corLWE. See Lemma 6.1 for details.

Lemma 4.3 (PVW is $(\frac{4r+1}{q})^\ell$-snake-eye resistant)**.** *For any $\lambda > 0$, $\ell = \mathsf{poly}(\lambda), q, p > 0$ and error distribution χ, let $\mathsf{aux} = (\ell, q, p, \chi)$, the PVW PKE scheme, is $(\frac{4r+1}{q})^\ell$-snake-eye resistant under $\mathsf{corLWE}_{n,\ell,w,q,\mathcal{U}^n,\chi}$ with corresponding parameters in Definition 3.3.*

Proof. Given a PPT adversary $\mathcal{A}$ that breaks the snake-eye resistance property of PVW, we construct an adversary that breaks $\mathsf{corLWE}_{n,\ell,w,q,\mathcal{U}^n,\chi}$ as follows.

Given $(A_1, \vec{b}_1, A_2, \vec{b}_2, \mathsf{sk}_1 - \mathsf{sk}_2)$, we directly feed $\mathsf{pk}_1 = (A_1, \vec{b}_1), \mathsf{pk}_2 = (A_2, \vec{b}_2)$ into $\mathcal{A}$. Upon receiving $(\vec{a}, \vec{b})$, if $|\vec{a}\ (\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j})| \leq 2r$ for all $j \in [\ell]$, output "the input is a valid LWE sample"; otherwise, output "the input is a random sample".

If the original $(A_1, \vec{b}_1), (A_2, \vec{b}_2)$ are valid LWE samples, it satisfies that $|\vec{a}\ \mathsf{sk}_{i,j} - \vec{b}_j| \leq r$ for $i \in \{1,2\}, j \in [\ell]$, and thus $|\vec{a}\ (\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j})| \leq 2r$ via triangle inequality. On the other hand, if the original $(A_1, \vec{b}_1), (A_2, \vec{b}_2)$ are random samples, $\mathcal{A}$ receives no information about $\mathsf{sk}_1, \mathsf{sk}_2$ (which are both uniformly randomly sampled). Thus, for $\vec{a} \neq \vec{0}$, we have $\Pr[\forall j \in [\ell], |\vec{a}\ (\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j})| \leq 2r] = (\frac{4r+1}{q})^\ell$.

Therefore, for any $\mathcal{A}$ with a noticeable advantage of breaking snake-eye resistance, the adversary we construct has the same advantage of breaking corLWE.

□

Remark 4.4. Our lemma is a slightly different version of Conjecture 3.6 introduced in [34]. The main differences are that (1) Conjecture 3.6 is with respect to Regev05 (which can be seen as a version of PVW with $\ell = 1$) instead of PVW, and (2) it conjectures $\delta = \frac{2r+1}{q}$. [34] further proved that for $(\frac{2r+1}{q})^{-\ell} = \mathsf{poly}(\lambda)$, PVW is $(\frac{2r+1}{q})^\ell$-snake-eye resistant under Conjecture 3.6 (see Lemma 3.7).

Our proof, instead, directly shows that PVW with arbitrary $\ell = \mathsf{poly}(\lambda)$ is $\delta = (\frac{4r+1}{q})^\ell$ snake-eye resistant. Thus, our Lemma 4.3 is more general (allows any polynomial size ℓ) but also has larger δ. However, since δ can be easily reduced by enlarging ℓ, this is not an issue in practice: one can simply enlarge ℓ and pad dummy plaintexts with little efficiency loss to achieve the desired δ (e.g., $\delta = \mathsf{negl}(\lambda)$; note that this is not achievable by [34] even under Conjecture 3.6).

Another small difference is that in [34], r can be arbitrary between 0 and $q/4$ but in our case r is a parameter output by PVW. Our proof can be extended to arbitrary r without any change, but for the generalized snake-eye resistance of PKE definition to make sense, we leave r for the underlying scheme to decide.

Together with Lemma 6.1 (showing that corLWE with uniform secrets is equivalent to regular LWE with uniform secrets), we resolve the open question stated in [34] asking whether the snake-eye resistance property for some widely-used lattice-based PKE scheme (such as Regev05 [49] and its variant PVW [47]) holds under the standard LWE assumption.

5 A More Efficient Snake-eye Resistant PKE

Although the PVW encryption scheme already satisfies the PKE snake-eye resistance defined in Definition 4.1, two major issues limit its efficiency. (1) Its public key size is large, $w\ell \log q = \omega(\ell n \log^2 q)$ (as to guarantee the scheme is CPA-secure and key-privacy, it needs to use leftover hash lemma and thus $w = \omega(n \log(q))$). (2) Since $w = \omega(n \log(q))$ is large, the error range r used for decryption needs to be large enough to satisfy $\Pr_{x_i \xleftarrow{\$} \chi_\sigma}[\sum_{i \in [w]} |x_i| > r] \leq \mathsf{negl}(\lambda)$: naturally, with a larger error range r, we need a larger ciphertext modulus q, which harms the efficiency of the PKE scheme.

Furthermore, in the application of OMR, the algorithm requires a homomorphic decryption of the underlying PKE scheme.[8] As discussed in [36], smaller r leads to more efficient homomorphic decryption circuits, which greatly boost the efficiency of the OMR construction.

5.1 Short-key LWE is not snake-eye resistant

Due to the two limitations above, the most widely-used lattice-based PKE schemes are based on short secret keys, such as the short-key LWE scheme (Definition 3.5) or its ring-variant [38], and other similar schemes, such as Crystal-Kyber [13]. For the short-key LWE scheme, the public key size is reduced to $n\ell \log(q)$ (asymptotically reduced by a factor of $\omega(\log(q))$ and practically > 10x smaller). Furthermore, by choosing r based on condition $\Pr_{x_i \xleftarrow{\$} \chi_\sigma}[\sum_{i\in[n]} |x_i| \cdot \beta > r] \leq \mathsf{negl}(\lambda)$, r could be much smaller if β is small (e.g., for ternary secrets, $\beta = 1$), practically > 10x smaller.

Thus, it is natural to ask whether such short-secret schemes are snake-eye-resistant. Unfortunately, we show that this is not the case. Specifically, we show that the short-key LWE PKE scheme and similar schemes are not snake-eye resistant by presenting the following attack.

Snake-eye attack on prior short-key LWE schemes. For simplicity, let us start with the binary secret key distribution, i.e. $\mathcal{D}_\beta = \mathcal{B}$, and $\ell = 1, q > r > n$. Then, the attacker directly outputs a ciphertext $(\vec{a}, b) = (\vec{1}, \lfloor n/2 \rceil)$. For *any* $\mathsf{sk} \leftarrow \mathcal{D}_\beta^n$, we have $|b - \langle \vec{a}, \mathsf{sk} \rangle| \leq \lfloor n/2 \rfloor$. Hence, this ciphertext decrypts to 0 for all honestly generated secret keys as long as $n/2 < r$. For schemes with parameters $r < n/2$, the attacker can instead set $\vec{a}$ to be $2r$ ones and $b = r$; thus, $|b - \vec{a}\,\mathsf{sk}| \leq r$.

Furthermore, this attack is easily generalizable: $(\vec{a} = \vec{2}, b = \lfloor n \rceil)$, or $(\vec{a} \xleftarrow{\$} [-\beta, \beta]^n, b = 0)$ for some small β, or even $(\vec{a} = q/2 \cdot \vec{1}, b = q/2)$ (which works for any even q instead of just $q \gg n$), and so on and so forth. With similar arguments, these wildcard ciphertexts can break the snake-eye resistance property. Therefore, it seems infeasible to rule out all possible wildcard ciphertexts to make the scheme snake-eye resistant.

Such attacks can all be extended to $\ell > 1$, as well as to Ring-LWE and Module-LWE, and to any short secrets distributions, which thus affects other LWE-based PKE schemes with short secrets (e.g., [38, 13]). In later sections, we also show that such attacks break the DoS-resistance of OMR constructions based on $\mathsf{shortLWE}$ (or similar schemes) like [36] and we implement a demo (see Section 8) and show that the attack indeed works for the construction in [36].

5.2 $\mathsf{LWEmongrass}$, New Snake-Eye Resistant PKE Construction

We propose a new lattice-based PKE construction that enjoys efficiency comparable to $\mathsf{shortLWE}$ (Definition 3.5) and is snake-eye resistant (Fig. 1 step (b)). Later shown in Section 7, this PKE scheme gives a new OMR construction that is both provably DoS-resistant and has similar efficiency to the state-of-the-art non-DoS-resistant OMR [36].

The attacks above fully rely on the fact that *all* secret key elements are short (with small norms). Therefore, our first key insight is to utilize a mix of short and uniform secrets. In other words, instead of solely sampling a short key or a uniform key, we sample both and then concatenate them together (i.e., $\mathsf{sk} \leftarrow \mathcal{D}_\beta^n || \mathcal{U}^k$ for some small β). For simplicity, let $\ell = 1$. The public key

[8] In more detail, OMR uses the PKE ciphertexts as clues and the detector needs to homomorphically decrypt these ciphertexts to determine who is the recipient of the message. See Section 7 for details.

$\mathsf{pk} = (A \xleftarrow{\$} \mathbb{Z}_q^{w\times(n+k)}, P = A\ \mathsf{sk} + \vec{e})$ where $\vec{e}$ is an error vector sampled from χ_σ and the ciphertext $\mathsf{ct} = (\vec{a}, b) \in \mathbb{Z}_q^{1\times(n+k)} \times \mathbb{Z}_q$. The intuition is that as long as the last k elements of $\vec{a}$ are non-zero, the snake-eye resistance property holds similarly to the PVW scheme.

However, similarly to $\mathsf{shortLWE}$, to make the public key size smaller (i.e., make $w \approx n$), the security proof of the scheme needs to use the LWE assumption instead of the leftover hash lemma. In other words, the encryption scheme samples a short key $\vec{x} \leftarrow \mathcal{D}_\beta^{1\times w}$ and computes $\mathsf{ct} = (\vec{x}A + \vec{e}_1, \vec{x}P + e_2 + t)$ where $\vec{e}_1, e_2$ are error vectors from χ_σ. The decryption algorithm computes $\vec{x}(A\ \mathsf{sk} + e) + e_2 + t - (\vec{x}A + \vec{e}_1)\mathsf{sk} = \vec{x}\ e + e_2 - \vec{e}_1\ \mathsf{sk} + t$ (where t is the encoding of the plaintext). However, since the last k elements of sk are uniformly sampled from $\mathbb{Z}_q$, the errors together can be as large as $q/2$, which breaks the correctness.

To fix this issue, the encryption procedure instead sets the last k elements of $\vec{e}_1$ to 0, i.e., the LWE sample generated by the encryption procedure has some error-free terms. In this case, even if the last k elements of the secret key are uniformly sampled, the error range r is bounded by only using β (i.e., the norm bound of the short part of the secret key), and thus guarantees correct decryption.

An immediate question is then whether the scheme is secure. To prove its security, we introduce another LWE variant *LWE with random hints (*rhLWE*)*. At a high-level, in rhLWE, we are given a standard LWE sample, and additionally some *random* linear combinations of the secret.

Definition 5.1 (LWE with Random Hints)**.** Let $n, \ell, w, q, \mathcal{D}, \chi$ be parameters dependent on λ, and let $k > 0$. The decisional LWE with random hints (rhLWE) assumption $\mathsf{rhLWE}_{n,\ell,w,q,\mathcal{D},\chi,k}$ states the following: any PPT adversary cannot distinguish $(A_1, A_1S + E, A_2, A_2S)$ and (A_1, B, A_2, A_2S) (except with negligible advantage), where $A_1 \xleftarrow{\$} \mathbb{Z}_q^{w\times n}$, $A_2 \xleftarrow{\$} \mathbb{Z}_q^{k\times n}$, $S \leftarrow \mathcal{D}^{n\times\ell}, E \leftarrow \chi^{w\times\ell}$, and $B \xleftarrow{\$} \mathbb{Z}_q^{w\times\ell}$.

In Lemma 6.1, we show that rhLWE with binary secret distribution is equivalent to standard LWE (for some small k), as needed to instantiate our scheme.

We formalize the new PKE construction with the above intuition which naturally extends to $\ell > 1$ in Algorithm 1. In Theorem 5.2, we show that our new construction is indeed a key-private PKE scheme, satisfying correctness, IND-CPA security, and key privacy, as defined in Definition 3.2. Finally, in Theorem 5.3 we show that it is snake-eye resistant (Definition 4.1).

Theorem 5.2. *For any $\lambda > 0$, $\ell = \mathsf{poly}(\lambda), q, p, k > 0$, secret distribution $\mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}$, and error distribution χ, let $\mathsf{aux} = (\ell, q, p, \mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}, \chi, k)$, $\mathsf{LWEmongrass}$ defined in Algorithm 1 is a key-private PKE scheme, under $\mathsf{rhLWE}_{n_2,1,n_1+\ell-k,q,\mathcal{D}_{\beta_2}^{n_2},\chi,k}$ and $\mathsf{LWE}_{n_1,\ell,n_2,q,\mathcal{D}_{\beta_1}^{n}||\mathcal{U}^k,\chi}$ where n_1, n_2 are chosen in Algorithm 1.*[9]

Proof. To prove that $\mathsf{LWEmongrass}$ is a key-private PKE scheme, we prove its correctness, CPA security, and key privacy, as follows.

- (Correctness) Correctness is straightforward: we start with $\ell = 1$. For any $m \in \mathcal{P}$, let $\mathsf{ct} = (\vec{a}, b) \leftarrow \mathsf{LWEmongrass.Enc}(\mathsf{pp}, \mathsf{pk}, m)$, $\vec{a}[n+1 : n_1] \neq \vec{0}$. Furthermore, $b - \vec{a}\ \mathsf{sk} = m \cdot \lfloor q/p \rfloor + \vec{x}P + e_2 - (\vec{x}A + \vec{e}_1)\mathsf{sk} = m \cdot \lfloor q/p \rfloor + \vec{x}(A\mathsf{sk} + E) + e_2 - \vec{x}A\mathsf{sk} + \vec{e}_1\mathsf{sk} = m \cdot \lfloor q/p \rfloor + \vec{x}E + e_2 + \vec{e}_1\mathsf{sk} = m \cdot \lfloor q/p \rfloor + e'$. Note that since the last k elements of $\vec{e}_1$ are zeros, the first n elements of sk are from $\mathcal{D}_{\beta_1}$, $\vec{x}$ are from $\mathcal{D}_{\beta_2}$, the first n elements of $\vec{e}_1$, and all the

[9]See Corollary 6.7 for $\mathsf{LWEmongrass}$'s security under standard LWE.

Algorithm 1 Snake-eye Resistant PKE $\mathsf{LWEmongrass}$

```
1: procedure $\mathsf{LWEmongrass.GenParam}(1^\lambda, \mathsf{aux} = (\ell, q, p, \mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}, \chi, k))$
2:     Set $p$ to be the plaintext modulus, $q$ to be the ciphertext modulus. Treat $\ell$ as the number of $\mathbb{Z}_p$ elements in plaintext, $\mathcal{D}_{\beta_1}$ as the secret key distribution, $\mathcal{D}_{\beta_2}$ as the distribution used in encryption, and $\chi$ as the error distribution.
3:     Set the minimum $n$ (let $n_1 = n + k$) and $n_2$ such that Theorems 5.2 and 5.3 assumptions hold. Let $k' = n_2 - n$.
4:     Set the minimum $r > 0$ s.t $\Pr_{x_i \xleftarrow{\$} \chi}[\sum_{i\in[n+n_2+1]} |x_i| \cdot \beta > r] \leq \mathsf{negl}(\lambda)$ for $\beta = \max(\beta_1, \beta_2)$
5:     If $r > \frac{q}{2p}$, output $\mathsf{pp} = \perp$.
6:     return $\mathsf{pp} = (n_1, n_2, \ell, q, p, \chi, r, \mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}, k)$
7: procedure $\mathsf{LWEmongrass.KeyGen}(\mathsf{pp})$          ▷ If $\mathsf{pp} = \perp$, output $\perp$ and same below.
8:     $\mathsf{sk} \leftarrow (\mathcal{D}_{\beta_1}^{\ell\times n} \| \mathcal{U}^{\ell\times k})^\intercal$
9:     $E \leftarrow \chi^{n_2\times\ell}$
10:    $A \xleftarrow{\$} \mathbb{Z}_q^{n_2\times n_1}$
11:    $\mathsf{pk} := (A, A\,\mathsf{sk} + E) = (A, P) \in \mathbb{Z}_q^{n_2\times n_1} \times \mathbb{Z}_q^{n_2\times\ell}$
12:    return $(\mathsf{pk}, \mathsf{sk})$
13: procedure $\mathsf{LWEmongrass.Enc}(\mathsf{pp}, \mathsf{pk}, \vec{m})$
14:    $\vec{x} \leftarrow \mathcal{D}_{\beta_2}^{1\times n_2}$
15:    If $(\vec{x}A)[n+1, n_1] = \vec{0}$, back to the previous step.
16:    $\vec{e}_1 \leftarrow \chi^{1\times n} \| 0^{1\times k}, \vec{e}_2 \leftarrow \chi^{1\times\ell}$
17:    $\vec{a} = \vec{x}A + \vec{e}_1, \vec{b} = \vec{x}P + \vec{e}_2 + \vec{m}\lfloor\frac{q}{p}\rfloor$
18:    return $(\vec{a}, \vec{b}) \in \mathbb{Z}_q^{1\times n_1} \times \mathbb{Z}_q^{1\times\ell}$
19: procedure $\mathsf{LWEmongrass.Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$
20:    If $\vec{a}[n+1 : n_1] = \vec{0}$, return $\perp$.
21:    compute $\vec{d} = \vec{a}\,\mathsf{sk} - \vec{b}$
22:    for $i \in [\ell]$ do
23:        If there exists $v \in \mathbb{Z}_p$ such that: $\left|\left\lfloor \vec{d}[i] \cdot \frac{p}{q} \right\rfloor - v\right| \leq r$, $\vec{m}[i] = v$
24:        otherwise, $\vec{m}[i] = \perp$
25:    return $\vec{m}$
```

$\vec{e}, e_2$ elements are from χ, $|e'| < r$ except with negligilble probability by line 4. Thus, the correctness holds. Furthermore, since $\ell = \mathsf{poly}(\lambda)$, this extends to $\ell > 1$ trivially.

- (CPA security) To prove CPA security, we define the following hybrids.

  hyb_0: same as $\mathsf{LWEmongrass}$.

  hyb_1: same as hyb_0 but during KeyGen, instead of computing P as in $\mathsf{LWEmongrass.KeyGen}$, sample it uniformly at random, i.e., $P \xleftarrow{\$} \mathbb{Z}_q^{n_2\times\ell}$.

  hyb_2: same as hyb_1 but during Enc, instead of computing $(\vec{a}, \vec{b})$ as in $\mathsf{LWEmongrass.KeyGen}$, sample it uniformly at random, i.e., $(\vec{a}, \vec{b}) \xleftarrow{\$} \mathbb{Z}_q^{1\times n_1} \times \mathbb{Z}_q^{1\times\ell}$ (if $\vec{a}[n+1, n_1] = \vec{0}$, resample) and then set $\vec{b} \leftarrow \vec{b} + \lfloor q/p \rfloor \cdot \vec{m}$.

  Note that hyb_2 trivially satisfies CPA security. hyb_1 and hyb_0 are indistinguishable by the hardness of $\mathsf{LWE}_{n_1,\ell,n_2,q,\mathcal{D}_\beta^n||\mathcal{U}^k,\chi}$. hyb_2 and hyb_1 are indistinguishable by the hardness of $\mathsf{rhLWE}_{n_2,1,n_1+\ell-k,q,\mathcal{D}_\beta^{n_2},\chi}$.

- (Key privacy) Key privacy can be proved the same way as CPA security, as hyb_2 trivially satisfies key privacy.

$\square$

We now show that our scheme is snake-eye resistant. Similarly to PVW, we rely on corLWE; we later show that the corLWE with the parametrization of interest is equivalent to LWE except for some small parameter loss.

Theorem 5.3 (($\frac{4r+1}{q})^\ell$-snake-eye Resistance of $\mathsf{LWEmongrass}$)**.** *For any $\lambda > 0$, $\ell = \mathsf{poly}(\lambda), q, p, k > 0$, secret distributions $\mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}$, and error distribution χ, let $\mathsf{aux} = (\ell, q, p, \mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}, \chi, k)$, $\mathsf{LWEmongrass}$ (Algorithm 1) is $\left(\frac{4r+1}{q}\right)^\ell$-snake-eye resistant, under $\mathsf{corLWE}_{n_1,\ell,n_2,q,\mathcal{D}_\beta^n||\mathcal{U}^k,\chi}$ with n_1, n_2, r in Algorithm 1.*[10]

Proof. The proof of $\mathsf{LWEmongrass}$ snake-eye resistance remains almost the same as for PVW in Lemma 4.3. The only change is that now $\vec{a}[n+1 : n_1] \neq \vec{0}$ instead of $\vec{a} \neq \vec{0}$ and our secret key has only the last k elements being uniform (using the notations of the proof in Lemma 4.3). Thus, we simply change the last sentence of the second last paragraph of the proof to "Thus, for $\vec{a}[n+1 : n_1] \neq \vec{0}$, we have $\Pr[\forall j \in [\ell], |\vec{a}(\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j})| \leq 2r] = (\frac{4r+1}{q})^\ell$". Everything else remains the same.

Therefore, similarly, for any $\mathcal{A}$ with a noticeable advantage, the adversary we construct has the same advantage breaking corLWE. $\square$

Remark 5.4. Note that our construction works for *any* $k > 0$ such that rhLWE holds. However, the larger k is, the easier the underlying rhLWE is, since we basically give out more random linear combinations of the secret as hints. Therefore, for the best security, we simply set $k = 1$. The only drawback in this case is that the sender needs to re-do the encryption with probability $\frac{1}{q}$ (see line 15). For most applications, this is not an issue as q is relatively large, and the rejection sampling takes little time. However, some applications might prefer larger k, and thus we present our scheme in its most general form.

Remark 5.5. One natural question is whether the scheme can instead be realized using Ring-LWE for better efficiency while maintaining the snake-eye resistance. Unfortunately, it seems hard: extending our scheme to Ring-LWE-based in the most natural way does not work.

The intuition is that encoding a plaintext of ℓ bits for most Ring-LWE-based encryption schemes simply puts the ℓ bits into the first ℓ coefficients of a ring element $m \in \mathcal{R}_t$ (ring elements are represented as a polynomial), where 1 is encoded as $t/2$ and 0 is encoded as 0. The decryption process for a ciphertext $a, b \in \mathcal{R}_t^2$ is $m' \leftarrow b - as$ (w.l.o.g., ignore the error term) for some honestly generated secret key s. Let S be the matrix representation of s (and let S_i denote the i-th column), $b[i]$ be the i-th coefficient of b, and $\vec{a}$ be the vector of all coefficients of a. Then, we have $m'[i] \leftarrow b[i] - \vec{a}S_i$. However, since S_i's are highly correlated, the adversary can craft some a and then predict the highly correlated $m'[i]$, even if s is unknown and has some uniform coefficients.

[10]See Corollary 6.7 for $\mathsf{LWEmongrass}$'s snake-eye resistance under standard LWE.

6 LWE Variants

In this section, we formally prove that the LWE variants proposed in the previous section are indeed equivalent to standard LWE for the parametrization needed to instantiate our scheme (Fig. 1 step (a)). Even though previous works have considered LWE variants with side information, either their formulations do not suit our needs or the reductions have larger parameter loss (e.g. they might consider a stronger variant). We define new LWE variants to achieve tighter reductions with better concrete efficiency and discuss in more detail their comparison with other LWE variants in Section 6.3.

6.1 LWE with Correlation

As in Definition 4.2, corLWE states that two LWE samples under two different LWE secrets are indistinguishable from random samples even given the difference between the two LWE secrets.

We prove that this variant is at least as hard as standard LWE (summarized in the upper segment of Fig. 1 step (a)) under three popular secret distributions: uniform secrets $\mathcal{U}$, binary secrets $\mathcal{B}$, and ternary secrets $\mathcal{T}$. Among all, $\mathcal{U}$ has essentially no security loss (except for w to $2w$ which is inherent since corLWE has two LWE samples each with size w). The binary secret distribution $\mathcal{B}$ has the largest parameter loss (but still relatively small). Intuitively, secret keys drawn from binary distribution have much less entropy than the ones drawn uniformly at random; therefore, their correlations leak a higher portion of their original randomness, thus suffering larger parameter loss.

Uniformly random secrets. For uniformly random secrets, there is almost no parameter loss when reducing LWE to corLWE, except for requiring the LWE sample to have $2w$ rather than w rows, which is intuitively the tightest scenario, as corLWE by definition includes two LWE samples, each with w components, under two different secrets. The idea behind the proof is straightforward: given a single LWE sample with size $2w$, we construct two LWE samples under two different uniformly sampled secret keys, each with size w by splitting the original sample into two and sampling a new uniform secret masking the original secret of one in the two samples. We formalize this intuition into the following lemma.

Lemma 6.1. *For any $n, \ell, w, q > 0$, and error distribution χ, let $\mathcal{U}_q$ denote the uniform distribution over q, it holds that $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{U}_q^n,\chi} \leq \mathsf{corLWE}_{n,\ell,w,q,\mathcal{U}_q^n,\chi}$.*

Proof. Given an adversary $\mathcal{A}$ that breaks $\mathsf{corLWE}_{n,\ell,w,q,\mathcal{U}_q^n,\chi}$, we construct the following adversary to break $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{U}_q^n,\chi}$.

For simplicity, we start with $\ell = 1$. Given LWE challenge $A \in \mathbb{Z}_q^{2w\times n}, \vec{b} \in \mathbb{Z}_q^{2w\times 1}$, divide $(A, \vec{b})$ into $(A_1, \vec{b}_1), (A_2, \vec{b}_2) \in \mathbb{Z}_q^{w\times n} \times \mathbb{Z}_q^{w\times 1}$. Sample $\mathsf{sk}' \leftarrow \mathcal{U}_q^n$. Compute $\vec{b}_2' \leftarrow \vec{b}_2 + A_2\mathsf{sk}'$, and feed $(A_1, \vec{b}_1, A_2, \vec{b}_2', -\mathsf{sk}')$ to $\mathcal{A}$; return the output of $\mathcal{A}$.

If the input challenge is an LWE sample with secret key sk, the constructed challenge is a valid corLWE sample: the first secret key is sk, the second secret key is $\mathsf{sk} + \mathsf{sk}'$, which is a freshly drawn secret key uniformly sampled from $\mathcal{U}_q$, and their difference is $-\mathsf{sk}'$. If the input challenge is a uniform random sample, then $\vec{b}_2'$ is also a uniformly sampled vector independent of $(A_1, A_2, \vec{b}_1, -\mathsf{sk}')$ as $\vec{b}_2$ is a uniformly sampled vector independent of these parts. The proof trivially extends to any $\ell > 1$. $\square$

Binary secrets. For the binary distribution, the proof is more involved as one cannot simply mask one uniform binary secret with another uniform binary secret key (the resulting secret key is no longer uniform binary). Thus, we devise new proof techniques.

Again, for simplicity, assume $\ell = 1$. The intuition behind this proof is as follows: the difference between the two secrets is a ternary vector $\vec{\beta} = \mathsf{sk}_1 - \mathsf{sk}_2 \in \{-1,0,1\}^n$. Then, if $\vec{\beta}[i] = -1$, it is obvious that $\mathsf{sk}_1[i] = 0, \mathsf{sk}_2[i] = 1$. Similarly, $\vec{\beta}[i] = 1$ means that $\mathsf{sk}_1[i] = 1$ and $\mathsf{sk}_2[i] = 0$. Therefore, these two scenarios fully leak the corresponding elements of the secrets. However, when $\vec{\beta}[i] = 0$, the only information leaked is that $\mathsf{sk}_1[i] = \mathsf{sk}_2[i]$, which still provides 1-bit of entropy, just like an element in a uniformly binary LWE secret. Note that $\Pr[\vec{\beta}[i] = 0] = 1/2$. Therefore, if the two secrets have length $\approx 2n$, the entropy of the two secrets conditioned on their difference is roughly the same as the entropy of one regular binary LWE secret with length n. To leverage such entropy equivalence, we embed the regular binary LWE secret with length n to two uniformly sampled binary secrets with length $n' > 2n$ (note that n' needs to be large enough so that there are at least n zeros in $\vec{\beta}$).

This results in a larger parameter loss: rhLWE has a secret length > 2x larger than the LWE secret length. Of course, this is still a relatively small security loss as it is only a small constant factor. To formalize this intuition, the following lemma and its corresponding proof are presented.

Lemma 6.2. *For any $n, w, q > 0$ and $\ell = \mathsf{poly}(n)$, and error distribution χ, for any constant $0 < \epsilon < 1$, let $n' = 2n/\epsilon$, it holds that $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{B}^n,\chi} \leq \mathsf{corLWE}_{n',\ell,w,q,\mathcal{B}^{n'},\chi}$.*

Proof. Given an adversary $\mathcal{A}$ that breaks $\mathsf{corLWE}_{n',\ell,w,q,\mathcal{B}^{n'},\chi}$, we construct the following adversary that breaks $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{B}^n,\chi}$.

For simplicity, we start with $\ell = 1$. Given LWE challenge $(A, \vec{b}) \in \mathbb{Z}_q^{2w\times n} \times \mathbb{Z}_q^{2w\times 1}$, sample two binary vectors $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow \mathcal{B}^{n'}$. Let $\vec{\beta} \leftarrow \mathsf{sk}_1 - \mathsf{sk}_2$, and let m denote the number of zeros in $\vec{\beta}$. If $m < n$, abort and return a random bit to the challenger.

We w.l.o.g. assume $\vec{\beta}[1:n]$ are zeros. (and discuss how to proceed if this is not the case in the next paragraph). Then, if $m \geq n$, divide $(A, \vec{b})$ into $(A_1, \vec{b}_1), (A_2, \vec{b}_2) \in \mathbb{Z}_q^{w\times n} \times \mathbb{Z}_q^{w\times 1}$. For $j \in [1,2]$, draw $A'_j \xleftarrow{\$} \mathbb{Z}_q^{w\times(n'-n)}$; let $A''_j := A_j || A'_j \in \mathbb{Z}_q^{w\times n'}$; let $\mathsf{sk}_j[1:n] \leftarrow 0^n$; compute $\vec{b}'_j \leftarrow \vec{b}_j + A''_j \mathsf{sk}_j$. Feed $(A''_1, \vec{b}'_1, A''_2, \vec{b}'_2, \vec{\beta})$ to $\mathcal{A}$. Return the output of $\mathcal{A}$.

If $\vec{\beta}[1:n] \neq 0^n$, simply permute $\mathsf{sk}_1, \mathsf{sk}_2, \vec{\beta}$ so that $\vec{\beta}[1:n] = 0^n$ before all these steps. Then, when feeding the challenge to $\mathcal{A}$, reverse this permutation to $\vec{\beta}$ and apply it to A''_1, A''_2 accordingly.

We first argue that this adversary aborts (and returns a random bit) with $\mathsf{negl}(n)$ probability. Note that $\vec{\beta}[i] = 0$ with probability $1/2$ (when $\mathsf{sk}_1[i] = \mathsf{sk}_2[i]$ which are both uniform binary elements). In expectation, $\vec{\beta}$ has $\mu = \frac{n'}{2} = \frac{n}{\epsilon}$ zeros and by Chernoff bound, $\Pr[m \leq n] = \Pr[m \leq \epsilon\mu] = \left(\frac{e^{1-1/\epsilon}}{\epsilon}\right)^n$. For constant ϵ, this happens with $\mathsf{negl}(n)$ probability.

If $(A, \vec{b})$ is an LWE sample with secret sk (unknown to our adversary), we argue that $(A''_1, \vec{b}'_1, A''_2, \vec{b}'_2, \vec{\beta})$ is a valid corLWE sample. By construction, $(A''_1, \vec{b}'_1)$ and $(A''_2, \vec{b}'_2)$ are valid LWE samples with keys $\mathsf{sk}'_1 := \mathsf{sk} || \mathsf{sk}_1[n+1:n']$ and $\mathsf{sk}'_2 := \mathsf{sk} || \mathsf{sk}_2[n+1:n']$, respectively, such that $\mathsf{sk}'_1 - \mathsf{sk}'_2 = \vec{\beta}$. So, it suffices to show that the keys sk'_1 and sk'_2 are uniform and independent binary vectors. Since $\mathsf{sk} \in \{0,1\}^n$ and for $j \in \{1,2\}$, $\mathsf{sk}_j \in \{0,1\}^{n'}$, the secrets sk'_1 and sk'_2 are binary. To show that sk'_1 and sk'_2 are independent, suppose we first sample independently sk_1 and sk_2 from $\mathcal{B}^n$, we select the first n positions where they share the same bit (which w.l.o.g. we assume to be the n first positions), and we replace those n bits with n independently randomly chosen bits; it is easy to see that the resulting two secrets are still independently sampled from $\mathcal{B}^{n'}$. This can be achieved as long as $\mathsf{sk}_1, \mathsf{sk}_2$ have at least n identical bits, which happens with $1 - \mathsf{negl}(n)$ as argued above. This is exactly what happens in the construction of the corLWE challenge, where the n bits used for replacement are the bits in sk.

If $(A, \vec{b})$ is uniform, the challenge we construct is a random corLWE challenge, since $\vec{b}''_1, \vec{b}''_1$ are masked by $\vec{b}_1, \vec{b}_2$ which are sampled uniformly at random.

Therefore, the adversary we construct has the same advantage as the advantage of $\mathcal{A}$ except with $\mathsf{negl}(n)$ probability.

When $\ell > 1$, since $\ell = \mathsf{poly}(n)$, by union bound, the argument above holds for all these ℓ columns of secrets except with $\ell \cdot \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$ probability. $\square$

Ternary secrets. For ternary secrets, using the same proof strategy (i.e., embed the LWE challenge secret into the positions with difference being zero) results in a parameter loss of > 3x (i.e., $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{T}^n,\chi} \leq \mathsf{corLWE}_{3n/\epsilon,\ell,w,q,\mathcal{T}^{3n/\epsilon},\chi}$ where $\mathcal{T}$ is the uniform ternary distribution). This seems quite loose, as intuitively ternary secrets should have less entropy loss compared to binary secrets when the difference between the two secrets is given.

To obtain a tighter bound, we employ a slightly different strategy. Observe that, unlike binary secrets, the difference of two ternary elements is in the set $\{-2, -1, 0, 1, 2\}$. In this case, only if $\vec{\beta}[i] \in \{-2, 2\}$, the secret elements are leaked fully, which happens with probability $\frac{2}{9}$. When $\vec{\beta}[i] = 0$ (which happens with probability 1/3), the entropy left is equivalent to a uniform ternary value, since again this difference only says that $\mathsf{sk}_1[i] = \mathsf{sk}_2[i]$. When $\vec{\beta}[i]$ is in $\{-1, 1\}$ (which happens with probability 4/9), the entropy left is equivalent to a uniform binary value: when $\vec{\beta}[i] = 1$, the two secret elements $(\mathsf{sk}_1[i], \mathsf{sk}_2[i])$ are either $(1, 0)$ or $(0, -1)$, and a similar argument holds for $\vec{\beta}[i] = -1$. Therefore, to fully utilize this observation, we reduce LWE with secrets sampled from a mix of uniform ternary and uniform binary elements (instead of a purely uniform ternary secret) to corLWE with uniform ternary secret. This results in an overhead of $\approx \frac{9}{7}$x instead of 3x, which greatly reduces the parameter loss. This intuition is formalized in the following lemma and proof.

Lemma 6.3. *For any $n, w, q > 0$ and $\ell = \mathsf{poly}(n)$, and error distribution χ, for any constant $0 < \epsilon < 1$, let $n' = \lceil \frac{9n}{7\epsilon} \rceil$, $n_t = \lceil \frac{3n}{7} \rceil, n_b = n - n_t$, $\mathcal{D} := \mathcal{T}^{n_t} || \mathcal{B}^{n_b}$, it holds that $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{D},\chi} \leq \mathsf{corLWE}_{n',\ell,w,q,\mathcal{T}^{n'},\chi}$.*

Proof. Given an adversary $\mathcal{A}$ that breaks $\mathsf{corLWE}_{n',\ell,w,q,\mathcal{T}^{n'},\chi}$, we construct the following adversary to break $\mathsf{LWE}_{n,\ell,2w,q,\mathcal{D},\chi}$.

For simplicity, we start with $\ell = 1$. Given LWE challenge $(A, \vec{b}) \in \mathbb{Z}_q^{2w \times n} \times \mathbb{Z}_q^{2w \times 1}$, sample two ternary vectors $\mathsf{sk}_1, \mathsf{sk}_2 \leftarrow \mathcal{T}^{n'}$. Let $\vec{\beta} \leftarrow \mathsf{sk}_1 - \mathsf{sk}_2$, m denote the number of zeros in $\vec{\beta}$, and m' denote the number of $\{\pm 1\}$ in $\vec{\beta}$. If $m < n_t$ or $m' < n_b$, abort and return a random bit to the challenger.

Again, we w.l.o.g assume that $\vec{\beta}[1 : n_t] = 0^{n_t}$, and $\vec{\beta}[n_t + 1 : n_t + n_b] \in \{\pm 1\}^{n_b}$. Divide $(A, \vec{b})$ into $(A_1, \vec{b}_1), (A_2, \vec{b}_2) \in \mathbb{Z}_q^{w \times n} \times \mathbb{Z}_q^{w \times 1}$. For $j \in$ [1,2]:

- draw $A'_j \xleftarrow{\$} \mathbb{Z}_q^{w \times (n'-n)}$
- let $A''_j := A_j || A'_j \in \mathbb{Z}_q^{w \times n'}$
- let $\mathsf{sk}_j[1 : n_t] \leftarrow 0^{n_t}$

- if $j = 1$, let $\mathsf{sk}_1[n_t + 1, n_t + n_b] = (\vec{\beta} - 1)/2$ (i.e., for $i \in [n_t + 1, n_t + n_b]$, $\mathsf{sk}_1[i] = 0$ if $\vec{\beta}[i] = 1$ and $\mathsf{sk}_1[i] = -1$ if $\vec{\beta}[i] = -1$)
- if $j = 2$, let $\mathsf{sk}_2[n_t + 1, n_t + n_b] = (-\vec{\beta} - 1)/2$ (i.e., for $i \in [n_t + 1, n_t + n_b]$, $\mathsf{sk}_2[i] = -1$ if $\vec{\beta}[i] = 1$ and $\mathsf{sk}_2[i] = 0$ if $\vec{\beta}[i] = -1$)
- compute $\vec{b}'_j \leftarrow \vec{b}_j + A''_j\ \mathsf{sk}_j$

Feed $(A''_1, \vec{b}'_1, A''_2, \vec{b}'_2, \vec{\beta})$ to $\mathcal{A}$. Return the output of $\mathcal{A}$. If $\vec{\beta}[1 : n_t] \neq 0^n$, or $\vec{\beta}[n_t+1 : n_t+n_b] \neq \{\pm 1\}$, simply permute $\mathsf{sk}_1, \mathsf{sk}_2, \vec{\beta}$ such that these two conditions are satisfied before all these steps. When feeding the challenge to $\mathcal{A}$, reverse this permutation to $\vec{\beta}$ and apply it to A''_1, A''_2 accordingly.

We first argue that this algorithm aborts with negligible probability. Similarly to the proof of Lemma 6.2, $\Pr[m < n_t] = \Pr[m < \lceil 3n/7 \rceil] \leq \left(\frac{e^{1-1/\epsilon}}{\epsilon}\right)^{3n/7} = \mathsf{negl}(n)$. Similarly, $\Pr[m' < n_b] = \mathsf{negl}(n)$. By union bound, the total probability of aborting is $\mathsf{negl}(n)$.

If the input is indeed an LWE sample with hidden secret sk, the challenge we construct is a valid corLWE sample. It suffices to show that the two secrets sk'_1 and sk'_2 of the LWE samples $(A''_1, \vec{b}''_1)$ and $(A''_2, \vec{b}''_2)$ are independently sampled from $\mathcal{T}^{n'}$. Assume that sk'_1 and sk'_2 are sampled as follows: first, sample $\mathsf{sk}_1, \mathsf{sk}_2$ from $\mathcal{T}^{n'}$; select the first n_t positions where $\mathsf{sk}_1[i] = \mathsf{sk}_2[i]$ and replace those n_t values with n_t independently randomly chosen ternary values; then, select the first n_b positions where $\mathsf{sk}_1[i] = \mathsf{sk}_2[i] \pm 1$; replace the positions where $\mathsf{sk}_1[i] = \mathsf{sk}_2[i] + 1$ with $(\mathsf{sk}'_1[i], \mathsf{sk}'_2[i]) := (\gamma, \gamma) + (0, -1)$ for some $\gamma \leftarrow \mathcal{B}$ (i.e., replace with the values (1,0) or $(0, -1)$ uniformly); replace the positions where $\mathsf{sk}_1[i] = \mathsf{sk}_2[i] - 1$ with $(\mathsf{sk}'_1[i], \mathsf{sk}'_2[i]) := (\gamma, \gamma) + (-1, 0)$ for some $\gamma \leftarrow \mathcal{B}$ (i.e., replace with the values $(-1$,0) or $(0, 1)$ uniformly). It is easy to see that the two resulting secrets are two independent vectors from the uniform ternary distribution $\mathcal{T}^{n'}$. This can be achieved as long as $\mathsf{sk}_1, \mathsf{sk}_2$ have at least n_t equal elements, and at least n_b elements differing by 1 or -1, which happens with $1 - \mathsf{negl}(n)$ as argued above. This is exactly what happens in the construction of the corLWE challenge, where the n_t values used for replacement are the ternary values of sk and the n_b values used for the replacement are the binary values of sk.

If the input is uniform, the challenge we construct is a random corLWE challenge since $\vec{b}''_1, \vec{b}''_2$ are masked by $\vec{b}_1, \vec{b}_2$ which are sampled uniformly at random.

Therefore, the adversary we construct has the same advantage as the advantage of $\mathcal{A}$ except with $\mathsf{negl}(n)$ probability. The proof also generalizes to $\ell = \mathsf{poly}(\lambda) > 1$ as the proof for Lemma 6.2. $\square$

Hybrid use of secret distributions. Lastly, we prove a general lemma saying that if LWE can be reduced to corLWE with secret distribution $\mathcal{D}'_1$, then it can be reduced to corLWE with secret distribution $\mathcal{D}'_1 || \mathcal{D}'_2$ for any (efficiently sample-able) distribution $\mathcal{D}'_2$. Essentially, it says that one can arbitrarily extend the secrets without hurting security. This is quite useful, as in our $\mathsf{LWEmongrass}$ (Algorithm 1), we concatenate a short secret and a uniform secret.

Lemma 6.4. *For any $n_1, n'_1, n'_2, w, q > 0$, $\ell = \mathsf{poly}(n)$, error distribution χ, and (efficiently sampleable) distributions $\mathcal{D}_1, \mathcal{D}'_1, \mathcal{D}'_2$, where vectors of length n_1, n'_1, n'_2 are sampled from $\mathcal{D}_1, \mathcal{D}'_1$ and $\mathcal{D}'_2$ respectively. If $\mathsf{LWE}_{n_1,\ell,2w,q,\mathcal{D}_1,\chi} \leq \mathsf{corLWE}_{n'_1,\ell,w,q,\mathcal{D}'_1,\chi}$, then it holds that $\mathsf{LWE}_{n_1,\ell,2w,q,\mathcal{D}_1,\chi} \leq \mathsf{corLWE}_{n'_1+n'_2,\ell,w,q,\mathcal{D}'_1||\mathcal{D}'_2,\chi}$.*

Proof. To prove this, we simply need to show that $\mathsf{corLWE}_{n'_1,\ell,w,q,\mathcal{D}'_1,\chi} \leq \mathsf{corLWE}_{n'_1+n'_2,\ell,w,q,\mathcal{D}'_1||\mathcal{D}'_2,\chi}$, which is straightforward. Given adversary $\mathcal{A}$ that breaks $\mathsf{corLWE}_{n'_1+n'_2,\ell,w,q,\mathcal{D}'_1||\mathcal{D}'_2,\chi}$, we construct the following adversary that breaks $\mathsf{corLWE}_{n'_1,\ell,w,q,\mathcal{D}'_1,\chi}$ as follows.

Given a $\mathsf{corLWE}_{n'_1,\ell,w,q,\mathcal{D}'_1,\chi}$ sample $(A_1, B_1, A_2, B_2, D = \mathsf{sk}_1 - \mathsf{sk}_2)$. Sample $A'_1, A'_2 \in \mathbb{Z}_q^{w\times n'_2}$, and $\mathsf{sk}'_1, \mathsf{sk}'_2 \leftarrow \mathcal{D}'^{\ell}_2$. Compute $B'_i \leftarrow B_i + A'_i\mathsf{sk}'_i$ for $i \in \{1,2\}$. Let $D' = \begin{pmatrix} D \\ \mathsf{sk}'_1 - \mathsf{sk}'_2 \end{pmatrix}$. Feed $(A_1||A'_1, B'_1, A_2||A'_2, B'_2, D')$ to $\mathcal{A}$. Return the output of $\mathcal{A}$.

If the input is a valid $\mathsf{corLWE}_{n'_1,\ell,w,q,\mathcal{D}'_1,\chi}$ sample, the input to $\mathcal{A}$ is trivially a $\mathsf{corLWE}_{n'_1+n'_2,\ell,w,q,\mathcal{D}'_1||\mathcal{D}'_2,\chi}$ sample. If the input is a random sample, the input to $\mathcal{A}$ is a random sample as B'_i is masked by uniformly random B_i. $\square$

6.2 LWE with Random Hints

To prove the CPA-security and key privacy of our new PKE construction (Algorithm 1), we relied on a new LWE variant rhLWE (Definition 5.1). Essentially, a rhLWE sample consists of a regular LWE sample, together with k random linear combinations of the secrets, which we call "hints". We provide a reduction from LWE with binary secrets to rhLWE with binary secrets (summarized in the bottom segment of Fig. 1 step (a)). Our proof is inspired by [5].

The intuition of our proof is as follows. For simplicity, we focus on the $\ell = 1$ case. Given an LWE sample $(A, \vec{b})$ with $A \in \mathbb{Z}_q^{(w+k)\times n}$ and $\vec{b} \in \mathbb{Z}_q^{w+k}$, we can guess the errors of the last k components with probability $(2|\chi|+1)^{-k}$, where $|\chi|$ is the norm bound of the error distribution χ (except with negligible probability).

If the input is a valid LWE sample and our guess is correct, we get a valid rhLWE sample. Otherwise, if the input is a uniform random sample or the guess is wrong, we argue that the input is of the form $(A_1, \vec{b}_1, A_2, \vec{b}_2 := A_2\vec{s})$, where $A_1 \xleftarrow{\$} \mathbb{Z}_q^{(w+k)\times n}$, $\vec{b}_1 \xleftarrow{\$} \mathbb{Z}_q^{w+k}$, $A_2 \xleftarrow{\$} \mathbb{Z}_q^{k\times n}$, and $\vec{s} \xleftarrow{\$} \{0,1\}^n$. This is done by proving a *reverse direction* (Lemma 6.6) of a simplified leftover hash lemma (i.e., $(\vec{x}A, A) \approx_s (\vec{u}, A)$ where x is uniform binary and A is uniform).

With the intuition above, for whatever advantage we have to break rhLWE, we can construct an adversary with $(2|\chi|+1)^{-k}$ smaller advantage. We formally capture the reduction in the following lemma.

Lemma 6.5 ($\mathsf{LWE} \leq \mathsf{rhLWE}$)**.** *For any $w > 0, \ell = \mathsf{poly}(n)$, any prime $q > 2$, distribution χ, $k > 0$ such that $(2|\chi|+1)^k = \mathsf{poly}(n)$, and $n = k \cdot \omega(\log(q))$, it holds that $\mathsf{LWE}_{n,\ell,w+k,q,\mathcal{B},\chi} \leq \mathsf{rhLWE}_{n,\ell,w,q,\mathcal{B},\chi,k}$.*

Proof. Given an addversary $\mathcal{A}$ that breaks $\mathsf{rhLWE}_{n,\ell,w,q,\mathcal{D},\chi,k}$, we construct the following adversary to break $\mathsf{LWE}_{n,\ell,w+k,q,\mathcal{D},\chi}$.

For simplicity, we start with $\ell = 1$. Let $w' = w+k$. Given LWE challenge $(A, \vec{b}) \in \mathbb{Z}_q^{w'\times n} \times \mathbb{Z}_q^{w'\times 1}$, sample $Y \xleftarrow{\$} \mathbb{Z}_q^{w'\times k}$, $\vec{r} \xleftarrow{\$} [-|\chi|, |\chi|]^{k\times 1}$. Let $Z = (0^{k\times w}||I_k) \in \mathbb{Z}_q^{k\times w'}$. Compute $T := YZ \pm I_{w'} \in \mathbb{Z}_q^{w'\times w'}$ such that T is invertible. Note that $\det(YZ + I_{w'}) = \det(YZ - I_{w'}) + 2$. Therefore, at least one of the two possible forms of T is invertible as $q > 2$ is a prime. W.l.o.g., assume $T := YZ + I_{w'}$ is invertible (the arguments below works similarly when $T := YZ - I_{w'}$ is invertible).

Let $A' \leftarrow TA, \vec{b}' \leftarrow T\vec{b} - Y\vec{r}$. Divide $A' = \begin{pmatrix} A'_1 \in \mathbb{Z}_q^{w\times n} \\ A'_2 \in \mathbb{Z}_q^{k\times n} \end{pmatrix}, \vec{b}' = \begin{pmatrix} \vec{b}'_1 \in \mathbb{Z}_q^{w\times 1} \\ \vec{b}'_2 \in \mathbb{Z}_q^{k\times 1} \end{pmatrix}$. Send $(A'_1, \vec{b}'_1, A'_2, \vec{b}'_2 - \vec{r})$ to $\mathcal{A}$ (replace $\vec{b}'_2 - \vec{r}$ with $\vec{b}'_2 + \vec{r}$ if $T := YZ - I_{w'}$ is invertible), and return whatever returned by $\mathcal{A}$.

Now we analyze the sample we send to $\mathcal{A}$. If the input is a valid LWE sample, we have

$$\begin{aligned}
\vec{b}' &= T\vec{b} - Y\vec{r} \\
&= T(A\vec{s} + \vec{e}) - Y\vec{r} \\
&= (TA)\vec{s} + T\vec{e} - Y\vec{r} \\
&= (TA)\vec{s} + I_{w'}\vec{e} + YZ\vec{e} - Y\vec{r} \\
&= (TA\vec{s}) + I_{w'}\vec{e} + Y\vec{e}[w+1 : w'] - Y\vec{r}
\end{aligned}$$

Therefore, as long as $\vec{e}[w+1 : w'] = \vec{r}$, then $Y\vec{e}[w+1 : w'] - Y\vec{r} = 0$, $\vec{b}' = A'\vec{s} + I_{w'}\vec{e}$, and thus $\vec{b}'_1 = A'_1\vec{s} + I_w\vec{e}[1 : w]$. Moreover, we would also have $\vec{b}'_2 - \vec{r} = \vec{b}'_2 + I_k\vec{e}[w+1 : w'] - \vec{r} = A'_2\vec{s}$, which becomes a noiseless hint as expected. Thus, $(A'_1, \vec{b}'_1, A'_2, \vec{b}'_2 - \vec{r})$ is a valid rhLWE sample. This happens with probability $1/((2|\chi| + 1)^k) = 1/\mathsf{poly}(n)$.

Then, we need to argue that if $\vec{e}[w+1 : w'] \neq \vec{r}$, we obtain a random sample. T is invertible and thus $A' = TA$ is uniformly at random. Then, since Y is uniformly at random, $Y(\vec{e}[w+1 : w'] - \vec{r})$ is trivially uniformly at random, and thus $\vec{b}'$ is uniformly at random. The only thing left to argue is that $(A'_2, b'_2 - \vec{r})$ is indistinguishable from $(A_2, A_2\mathsf{sk})$ where $A_2 \xleftarrow{\$} \mathbb{Z}_q^{k\times n}, \mathsf{sk} \leftarrow \mathcal{B}^n$. By Lemma 6.6, this holds with probability $1 - 2^{n-k\log(q)} = 1 - \mathsf{negl}(n)$, $n = k \cdot \omega(\log(q))$.

Lastly, we just need to argue that if the input is a random sample, the sample we feed to $\mathcal{A}$ is also a random sample: T is invertible and thus $TA, T\vec{b}$ are both uniformly at random. Thus, $A'_1, \vec{b}'_1$ are both uniformly at random. Similar to before, $(A'_2, \vec{b}'_2 - \vec{r})$ is a valid hint with respect to some uniform binary secret by Lemma 6.6.

Thus, if the input is a valid LWE sample, the adversary constructed above has an advantage of $\epsilon/(2|\chi| + 1)^k$ where ϵ is the advantage of the original adversary $\mathcal{A}$ breaking rhLWE.

Lastly, we discuss how to extend to $\ell > 1$. Given an rhLWE with $\ell > 1$, we create a hybrid that replaces one of the ℓ components with uniform random samples and this hybrid is indistinguishable from the original sample as otherwise rhLWE with $\ell = 1$ is broken. The hybrid argument is repeated for ℓ times with a loss of ℓ factor in terms of security. However, since $\ell = \mathsf{poly}(n)$, this is acceptable. $\square$

Lemma 6.6 (Simplified Reverse-LHL)**.** *For any prime $p > 0, k > 0, k' = k\omega(\log q)$, let $H(A, \vec{u})$ be defined as follows: given $A \in \mathbb{Z}_q^{k\times k'}$ and $\vec{u} \in \mathbb{Z}_q^{k\times 1}$, define $\mathcal{X}_u := \{\vec{x}^* \mid A\vec{x}^* = \vec{u}, \vec{x}^* \in \{0,1\}^{k'}\}$; if $\mathcal{X}_u = \emptyset$, output $(2, 2, \dots, 2)$ (i.e., a non-binary vector); otherwise, output $\vec{x} \xleftarrow{\$} \mathcal{X}_u$. It holds that the statistical distance between $(H(A, \vec{u}), A)$ and $(\vec{y}, A)$ is $O(2^{-k'+k\log(q)})$, where $A \xleftarrow{\$} \mathbb{Z}_q^{k\times k'}, \vec{u} \xleftarrow{\$} \mathbb{Z}_q^{k\times 1}, \vec{y} \xleftarrow{\$} \{0,1\}^{k'}$.*

Proof. Let $U := \{A\vec{x}, \forall \vec{x} \in \{0,1\}^{k'}\}$ and $P := \Pr_{\vec{u}\xleftarrow{\$}\mathbb{Z}_q^{k\times 1}}[H(A,\vec{u}) \notin \{0,1\}^{k'}] = \sum_{\mathbb{Z}_q^k \setminus U} \frac{1}{q^k}$ (i.e., the probability that $\mathcal{X}_u = \emptyset$). By definition, the statistical distance between $(H(A, \vec{u}), A)$ and $(\vec{y}, A)$ is computed as $Q + P$ where Q is defined as follows:

$$\begin{aligned}
Q &:= \sum_{\vec{x}\in\{0,1\}^{k'}} \left| \Pr_{\vec{u}\xleftarrow{\$}\mathbb{Z}_q^{k\times 1}} [H(A,\vec{u}) = \vec{x}] - \frac{1}{2^{k'}} \right| \\
&= \sum_{\vec{u}'\in U} \left(\sum_{\vec{x}\in\mathcal{X}_{u'}} \left| \Pr_{\vec{u}\xleftarrow{\$}\mathbb{Z}_q^{k\times 1}} [H(A,\vec{u}) = \vec{x}] - \frac{1}{2^{k'}} \right| \right) \\
&= \sum_{\vec{u}'\in U} \left(\sum_{\vec{x}\in\mathcal{X}_{u'}} \left| \frac{1}{|\mathcal{X}_{u'}|} \cdot \frac{1}{q^k} - \frac{1}{2^{k'}} \right| \right) \\
&= \sum_{\vec{u}'\in U} \left| \frac{1}{q^k} - \frac{|\mathcal{X}_{u'}|}{2^{k'}} \right|
\end{aligned}$$

From step one to two, we simply rearrange the summations from a summation over all possible $\vec{x}$ to a summation over all the solutions of $A\vec{x}$ from all possible values of $A\vec{x}$ (which are equivalent). From step two to three, since $\vec{u}$ is uniformly drawn and thus the probability of selecting exactly the $\vec{u} = \vec{u}'$ is q^{-k}; and then for all the possible solutions, the probability of selecting $\vec{x}$ is $1/|\mathcal{X}_{u'}|$ and these two events are independent. The last step is a formula simplification.

Now, LHL states that let $A \xleftarrow{\$} \mathbb{Z}_q^{k\times k'}$, $\vec{x} \xleftarrow{\$} \{0,1\}^{k'}$, and $\vec{u} \xleftarrow{\$} \mathbb{Z}_q^k$, it holds that the statistical distance between $(A\vec{x}, A)$ and $(A, \vec{u})$ is $O(2^{-k'+k})$, which means

$$\begin{aligned}
&\sum_{\vec{u}\in\mathbb{Z}_q^k} \left| \Pr_{\vec{x}\xleftarrow{\$}\{0,1\}^k} [A\vec{x} = \vec{u}] - \frac{1}{q^k} \right| \\
&= \sum_{\vec{u}\in\mathbb{Z}_q^k} \left| \frac{|\mathcal{X}_u|}{2^{k'}} - \frac{1}{q^k} \right| \\
&= \sum_{\vec{u}'\in U} \left| \frac{|\mathcal{X}_{u'}|}{2^{k'}} - \frac{1}{q^k} \right| + \sum_{\mathbb{Z}_q^k\setminus U} \left| 0 - \frac{1}{q^k} \right| \\
&= Q + P = O(2^{-k'+k\log(q)})
\end{aligned}$$

□

Adversarially chosen A_2. Note that above, the hints of the secrets (i.e., A_2) are randomly chosen, which is already enough to prove the security of our construction. However, an *independently* interesting question is to consider whether a similar result would still hold under a maliciously chosen A_2. The answer is yes but only under $q^k = \mathsf{poly}(n)$ instead of just $(2|\chi| + 1)^k = \mathsf{poly}(n)$. A similar result and proof are in [5], and we omit the details here. Our reduction is tighter tailored to our LWE variant in terms of both theoretical result and concrete parameter choices.

LWEmongrass. With all these (Theorems 5.2 and 5.3 and Lemmas 6.3 and 6.5), we conclude that LWEmongrass is secure and snake-eye resistant under *standard* LWE assumptions, formalized as the following corollary.

Corollary 6.7. *For any $\lambda > 0, \ell = \mathsf{poly}(\lambda)$, $q, p > 0$, secret distributions $\mathcal{D}_{\beta_1} = \mathcal{T}, \mathcal{D}_{\beta_2} = \mathcal{B}$, error distribution χ, and any $k > 0$ such that $(2|\chi| + 1)^k = \mathsf{poly}(\lambda)$, let $\mathsf{aux} = (\ell, q, p, \mathcal{D}_{\beta_1}, \mathcal{D}_{\beta_2}, \chi, k)$, $\mathsf{LWEmongrass}$ is a key-private PKE scheme under $\mathsf{LWE}_{n_2,1,n_1+\ell,q,\mathcal{T}^{n_2},\chi}$ and $\mathsf{LWE}_{n_1,\ell,n_2,q,\mathcal{T}^n||\mathcal{U}^k,\chi}$ for $n = n_1-k$; furthermore, $\mathsf{LWEmongrass}$ is $(4r+1)^\ell$-snake-eye resistant, under $\mathsf{LWE}_{n_t+n_b,\ell,2\cdot n_2,q,\mathcal{T}^{n_t}||\mathcal{B}^{n_b},\chi}$, where let n' be the minimum such that $n = \lceil \frac{9n'}{7\epsilon} \rceil$, then $n_t = \lceil \frac{3n'}{7} \rceil, n_b = n' - n_t$; for any constant $\epsilon > 0$ and n_1, n_2, r chosen in Algorithm 1.*

6.3 Discussion of other LWE variants with Side-information

The LWE variants we propose are both LWE with side information. They share similarities with LWE variants with side information introduced in prior works. However, those variants and reductions do not work well for our purposes, as we discuss below.

ExtLWE. In the extended-LWE problem (ExtLWE) [44, 5, 17, 3], we are given a regular LWE sample together with $\langle \vec{r}, \vec{e} \rangle + \vec{e}\,'$ for an adversarially chosen vector $\vec{r}$, where $\vec{e}$ is the error of the LWE sample and $\vec{e}\,'$ is a newly sampled error. In other words, we are given a noisy linear combination of the LWE error. [5, 17] remove the extra error term $\vec{e}\,'$ and show that LWE with *uniform secrets* can be reduced to ExtLWE with $\vec{e}\,' = 0$. Our proof for Lemma 6.5 is mostly inspired by the proof in [5], except that our reduction is tighter given for our new LWE variant.

HintLWE. The Hint-LWE problem (HintLWE) [33, 18, 32] gives out $\vec{e} + \vec{e}\,'$ (instead of a noisy inner product as in ExtLWE) together with the LWE sample. As far as we know, all reductions from LWE to HintLWE require $\vec{e}\,'$ to be non-zero, and thus do not suit our needs.

EntLWE. The Entropic LWE problem (EntLWE) [28, 27, 16] focuses on how LWE with uniform secrets can be reduced to LWE with secrets from a distribution with certain min-entropy. Therefore, the LWE assumption as defined in Definition 3.1 can be seen as the EntLWE assumption, since we allow LWE to be parametrized by different secret distributions rather than specifically the uniform distribution. However, we only focus on uniform, binary, and ternary cases, which are all widely used secret distributions, and thus to avoid extra complexity, we introduce Definition 3.1 as the standard LWE assumption instead of EntLWE.

7 Constructing OMR from Snake-eye Resistant PKE

The following recalls model and definition of the Oblivious Message Retrieval and its Denial-of-Service resistance.

7.1 OMR models

Overview of OMR system model. As a high-level overview, an OMR protocol works in an anonymous message delivery system. A *sender* wants to send a message to a *recipient*, without a direct secure channel or revealing the information about the recipient. So the sender broadcasts the message (e.g., by putting it on a *bulletin board*). The recipient wants to retrieve all the messages addressed to it (denoted as the *pertinent* messages) by outsourcing this task to a *detector* but without revealing any information about the pertinent messages.

The workflow is as follows. The recipient publishes a *clue key* (used for message generation) and a *detection key* (used for retrieval). The sender uses the clue key of a recipient to generate a *clue* which together with a payload constitutes a message to be put on the *bulletin board*. The detector uses the detection key of the recipient together with the board to generate a *digest* including all the pertinent messages for the recipient. The recipient, who receives the digest from the detector, uses its secret key to decrypt the digest, and recovers all the pertinent messages.

DoS attack on OMR. In a Denial-of-Service (DoS) attack on OMR, the adversary tries to craft a *single* clue that could be detected as "pertinent" by *many* recipients, with the extra communication and processing this entails.

In particular, such attacks can cause completeness failure due to overflow, as follows. For efficiency, the digest size should be much smaller than the board size. Ideally, its size should only be proportional to the number of pertinent messages. However, the recipient itself does not know how many messages are pertinent without scanning the board. Therefore, the recipient provides some bound $\bar{\gamma}$ on the number of pertinent messages. If the real number of pertinent messages $\gamma > \bar{\gamma}$, OMR protocol returns "overflow" instead of the pertinent messages. By the nature of the system model, an attacker can overflow a *single* recipient by sending a lot of dummy messages (and paying the cost incurred by sending), such that the recipient would under-estimate γ by sending a small $\bar{\gamma}$. However, the DoS attack in this threat model is much worse: the attacker might overflow *many* recipients at the same time. Moreover, even if the recipient knows the exact γ and sends a large enough $\bar{\gamma}$, the runtime of the detector and the digest size are both dependent on $\bar{\gamma}$ and thus can greatly affect the efficiency of the scheme for *many* recipients. A DoS-resistant OMR aims to prevent such an attack.

A DoS attack might also be done by a malicious *recipient* who drafts a malicious clue key, such that any honestly generated clue using that clue key ends up (unknowingly) creating a clue that is detected as partinent by many recipients. Thus, we need to prevent such attacks from both malicious senders and recipients.

Notice that such an attack is indeed a concern: as shown in [34, Sec 8], its original construction (without its patch) is vulnerable and similar works like [7, 39] are all vulnerable (see [34, Section 2] for details). Furthermore, as we show in Section 5, PerfOMR is also vulnerable.

The threat model. With these in mind, we specify the threat model of the DoS OMR, which is the same as in prior works that considered DoS attacks on OMR [34, 35].

Privacy. We consider an adversary that wishes to learn metadata about which messages are addressed to which recipient, and about the identity of recipients that perform message-retrieval queries. The adversary can read all public information (including all board messages and all public keys), and all communication between detectors and the recipients. The adversary may corrupt all the parties in the system, except for the message recipients whose privacy is to be protected and the senders of those messages. The adversary may behave maliciously and send malformed messages and keys on behalf of all corrupted parties; but it is computationally bounded (i.e., cannot break the underlying computational assumptions).

Integrity. We consider an adversary that wishes to prevent the recipients from receiving its pertinent messages. For this, we assume the detector is honest-but-curious, but the recipients and the senders may be fully malicious as above (and thus are allowed to perform the DoS attacks specified above.).

7.2 DoS-resistant OMR Definition

OMR in DoS model is formally defined as follows, adapted from the definition in [34], with the following five PPT algorithms. At a high level, the system has a GenParams to generate the general parameters; each recipient uses KeyGen to generate their secret keys, clue keys, and detection keys; each sender then uses $\mathsf{GenClue}$ together with the intended recipient's clue key to generate a clue. To perform a retrieval, the detector uses $\mathsf{Retrieve}$ together with the recipient's detection key (and all the messages on the bulletin board) to generate a digest. After receiving the digest, the recipients use their secret key to Decode and obtain their pertinent messages.

- $\mathsf{pp} \leftarrow \mathsf{GenParams}(1^\lambda, \epsilon_\mathrm{p}, \epsilon_\mathrm{n})$: takes a security parameter λ, a false positive rate ϵ_p and a false negative rate ϵ_n; outputs the public parameter pp.
- $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_\mathsf{clue}, \mathsf{pk}_\mathsf{det})) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$: outputs a secret key sk and a public key pk consisting of a clue key $\mathsf{pk}_\mathsf{clue}$ and a detection key pk_det.
- $c \leftarrow \mathsf{GenClue}(\mathsf{pp}, \mathsf{pk}_\mathsf{clue}, x)$: takes a clue key and a payload $x \in \mathcal{P}$; outputs a clue $c \in \mathcal{C}$.
- $M \leftarrow \mathsf{Retrieve}(\mathsf{pp}, \mathsf{BB}, \mathsf{pk}_\mathsf{det}, \bar{\gamma})$: takes as input a board $\mathsf{BB} = \{(x_1,c_1), \ldots, (x_N,c_N)\}$ for some size N, a detection key pk_det, and an upper bound $\bar{\gamma}$ on the number of pertinent messages addressed to that recipient; outputs a digest M.
- $\mathsf{PL} \leftarrow \mathsf{Decode}(\mathsf{pp}, M, \mathsf{sk})$: takes the digest M and corresponding secret key sk; outputs either a decoded payload list $\mathsf{PL} \subset \mathcal{P}^k$ or an overflow indication $\mathsf{PL} = \mathsf{overflow}$.

To define the completeness and soundness, we first define an indicator $\mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_\mathsf{clue}, \mathsf{sk})$, which serves as the ground truth for whether a given clue c is pertinent to a given user with respect to its clue key $\mathsf{pk}_\mathsf{clue}$ and/or its secret key sk. For honestly generated clues, the indicator should give the natural answer (output 1 for the key used to generate the clue except with ϵ_n probability). For otherwise-generated clues, the indicator may answer arbitrarily, except that it must still make up its mind, i.e., not claim more than one honest recipient as the intended one, except with ϵ_p probability. This *collision resistance* property means that, while a malicious sender can (inevitably) craft a message that is considered pertinent by one user, it is difficult for it to spam *multiple* users with a *single* message. The OMR DoS-completeness and DoS-soundness are then defined against this indicator.

Definition 7.1 (DoS-resistant OMR, [34]). Let OMR be a scheme with the five PPT algorithms above with input security parameter λ and error rates $\epsilon_\mathrm{n}, \epsilon_\mathrm{p}$. An *indicator* for OMR is a function $b \leftarrow \mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_\mathsf{clue}, \mathsf{sk})$ on a public parameter pp, a message (x, c), a clue key $\mathsf{pk}_\mathsf{clue}$, and a secret key sk, that outputs $b \in \{0,1\}$, such that:

- *(Indicator completeness)* For $\mathsf{pp} \leftarrow \mathsf{GenParams}(1^\lambda, \epsilon_\mathrm{p}, \epsilon_\mathrm{n})$, honest-generated key pair $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_\mathsf{clue},\cdot)) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, for any payload x and honestly-generated clue $c \leftarrow \mathsf{GenClue}(\mathsf{pp}, \mathsf{pk}_\mathsf{clue}, x)$, it holds that:

$$\Pr[\mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_\mathsf{clue}, \mathsf{sk}) = 1] \geq 1 - \epsilon_\mathrm{n} - \mathsf{negl}(\lambda) \ .$$

- *(Collision resistance)* For any PPT adversary $\mathcal{A}$, let $\mathsf{pp} \leftarrow \mathsf{GenParams}(1^\lambda, \epsilon_\mathrm{p}, \epsilon_\mathrm{n})$, two honest-generated key pairs $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_\mathsf{clue}, \cdot)) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$ and $(\mathsf{sk}', \mathsf{pk}' = (\mathsf{pk}'_\mathsf{clue}, \cdot)) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, and adversarially-generated $(x,c) \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{pk}')$, for $b \leftarrow \mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_\mathsf{clue}, \mathsf{sk})$ and $b' \leftarrow \mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}'_\mathsf{clue}, \mathsf{sk}')$, it holds that:

$$\Pr[b = 1 \ \wedge \ b' = 1] \leq \epsilon_\mathrm{p} + \mathsf{negl}(\lambda) \ .$$

An OMR scheme OMR is *DoS-resistant* for ϵ_{n} and ϵ_{p} if there exists an indicator $\mathcal{I}$ with an indicator false negative rate ϵ_{in} for OMR such that for any PPT adversary $\mathcal{A}$, for $\mathsf{pp} \leftarrow \mathsf{GenParams}(1^\lambda, \epsilon_{\mathrm{p}}, \epsilon_{\mathrm{n}})$, $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_{\mathsf{clue}}, \mathsf{pk}_{\mathsf{det}})) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, and adversarially-generated board $\mathsf{BB} \leftarrow \mathcal{A}(\mathsf{pp},\mathsf{pk})$ where $\mathsf{BB} = ((x_1,c_1), \ldots, (x_N,c_N))$, $(x_i)_{i\in[N]}$ all unique, for any $0 < \bar{\gamma} \leq N$, let $M \leftarrow \mathsf{Retrieve}(\mathsf{pp}, \mathsf{BB}, \mathsf{pk}_{\mathsf{det}}, \bar{\gamma})$, $\mathsf{PL} \leftarrow \mathsf{Decode}(M, \mathsf{sk})$:

- *(DoS-completeness)* Let $\gamma = \sum_{i=1}^{N} \mathcal{I}(\mathsf{pp}, x_i, c_i, \mathsf{pk}_{\mathsf{clue}}, \mathsf{sk})$. Then either $(\gamma > \bar{\gamma}) \wedge (\mathsf{PL} = \mathsf{overflow})$, or $\Pr[x_j \in \mathsf{PL} \mid \mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_{\mathsf{clue}}, \mathsf{sk}) = 1] \geq 1 - \mathsf{negl}(\lambda)$ for all $j \in [N]$.
- *(DoS-soundness)* $\Pr[x_j \in \mathsf{PL} \mid \mathcal{I}(\mathsf{pp}, x, c, \mathsf{pk}_{\mathsf{clue}}, \mathsf{sk}) = 0] \leq \mathsf{negl}(\lambda)$ for all $j \in [N]$.

Furthermore, it satisfies the following computational security property:

- (Computational privacy) For any PPT adversary $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$: let $\mathsf{pp} \leftarrow \mathsf{GenParams}(1^\lambda, \epsilon_{\mathrm{p}}, \epsilon_{\mathrm{n}})$, $(\mathsf{sk}, \mathsf{pk} = (\mathsf{pk}_{\mathsf{clue}}, \cdot)) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$ and $(\mathsf{sk}', \mathsf{pk}' = (\mathsf{pk}'_{\mathsf{clue}},\cdot)) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$. Let the adversary choose a payload x and remember its state: $(x,\mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk}, \mathsf{pk}')$. Let $c \leftarrow \mathsf{GenClue}(\mathsf{pp}, \mathsf{pk}_{\mathsf{clue}}, x)$ and $c' \leftarrow \mathsf{GenClue}(\mathsf{pp}, \mathsf{pk}'_{\mathsf{clue}}, x)$, then $|\Pr[\mathcal{A}_2(\mathsf{st}, c) = 1] - \Pr[\mathcal{A}_2(\mathsf{st}, c') = 1]| \leq \mathsf{negl}(\lambda)$.

We omit two properties for simplicity. The first is compactness, which is an efficiency requirement on the digest size $|M| \ll |\mathsf{BB}|$. The second one is a security notion called key-unlinkability, which essentially says that a recipient can generate multiple clue keys and detection keys using the same secret key while all these keys are unlinkable. For both properties, we achieve exactly the same guarantee as in what is achieved in [34, 36] since our construction is based on the same framework used in [34] and these properties are orthogonal to the main point of the paper. See Remark 7.3 for details.

7.3 DoS-resistance of OMRp2 in [34]

Using Lemma 4.3, we prove that the OMR construction in [34] is indeed DoS-resistant as conjectured. Here we first recall how $\mathsf{OMRp2}$ ([34, Alg.8]) is constructed, which also serves as the general framework we use to construct our own OMR.

OMRp2 (OMR with PVW). The construction $\mathsf{OMRp2}$ is proposed in [34], which is based on PVW encryption (Definition 3.3) and BFV homomorphic encryption (Section 3.6).

Each recipient first generates a pair of PVW keys $(\mathsf{pk}_{\mathsf{PVW}}, \mathsf{sk}_{\mathsf{PVW}})$ and a pair of BFV keys $(\mathsf{pk}_{\mathsf{BFV}}, \mathsf{sk}_{\mathsf{BFV}})$. The recipient publishes $\mathsf{pk}_{\mathsf{clue}} = \mathsf{pk}_{\mathsf{PVW}}$ as the clue key and the sender uses $\mathsf{pk}_{\mathsf{clue}}$ to encrypt $\vec{0} \in \mathbb{Z}_2^\ell$ into a ciphertext $(\vec{a}, \vec{b})$. The ciphertext, serving as the clue, is published with the corresponding payload onto the bulletin board. Thus, the PVW scheme (or more generally, the underlying PKE scheme) is used to implicitly convey the pertinency of a message on the public bulletin board. To retrieve the pertinent messages, a recipient encrypts its own PVW secret key $\mathsf{sk}_{\mathsf{PVW}}$ under $\mathsf{pk}_{\mathsf{BFV}}$, denoted as $\mathsf{ct}_{\mathsf{sk}}$, and sends $\mathsf{pk}_{\mathsf{det}} = (\mathsf{pk}_{\mathsf{BFV}}, \mathsf{ct}_{\mathsf{sk}})$ as the detection key to the detector. The detector uses $\mathsf{pk}_{\mathsf{det}}$ to homomorphically decrypt all clues into ℓ bits (the design of this homomorphic circuit will be discussed later), and then homomorphically negates all the ℓ bits and multiplies them.

After this step, with N clues on the bulletin board, the detector will get N bits. All the pertinent messages have the corresponding bits being 1 (except with negl probability) and the others being

0 except with δ probability (recall that PVW is δ-snake-eye resistant). For simplicity, consider $\delta = \mathsf{negl}(\lambda)$.

With this (sparse) bit vector of size N together with the N payloads in BB, the detector generates a digest containing only the payloads of the messages with the corresponding bits being 1, by homomorphically evaluating an encoding scheme. The recipient, after receiving the digest back, decrypts the digest, and performs a decoding procedure to obtain all the payloads.

We skip the details of this encoding procedure on the detector side and the decoding scheme on the recipient side since they are irrelevant to our work and refer readers to [34] for details.

DoS resistance of OMRp2. [34] shows that the construction OMRp2 is DoS-resistant under the conjecture that PVW is snake-eye resistant with $\delta = ((2r+1)/q)^{\ell}$ ([34, Thm 9.3]). In Lemma 4.3, we prove that PVW is indeed snake-eye resistant with $\delta = ((4r+1)/q)^{\ell}$. Therefore, the original OMR scheme given in [34] is now formally proven to be DoS-resistant with a false positive rate $\epsilon_{\mathrm{p}} = ((4r+1)/q)^{\ell}$ (instead of $((2r+1)/q)^{\ell}$ as originally conjectured); for a specific ϵ_{p}, this means that ℓ needs to be slightly increased.

7.4 Non-DoS Resistance of PerfOMR in [36]

PerfOMR (OMR with shortLWE). Following the aforementioned framework, [36] proposes shortLWE (Definition 3.5) inplace of PVW and constructs PerfOMR.[11] This gives two advantages: (1) the clue key size is now greatly reduced due to a smaller PKE public key; (2) the detector runtime is faster as r is greatly reduced and thus a more efficient homomorphic decryption circuit can be designed.

The DoS attack against PerfOMR. Unfortunately, as shown in Section 5.1, the PKE scheme shortLWE is not snake-eye resistant. Hence, the resulting OMR scheme PerfOMR ([36, Alg.4&5]) is not DoS-resistant: the attacker can generate the ciphertext we propose in the attack as the clue, and then many, if not all, honest recipients will detect it as pertinent. In Section 8, we implement and demonstrate that such attacks indeed work as expected.

As mentioned at the beginning of this section, such an attack means that an adversary can generate γ clues, which will be detected as pertinent by a large number of recipients. Unless these recipients set $\bar{\gamma} > \gamma$, the decoding protocol after retrieval outputs "overflow". However, if $\bar{\gamma}$ is set to be too large, since the detector computation time and the communication cost are dependent on $\bar{\gamma}$ (polylog and linear dependency respectively), the efficiency is greatly affected.

7.5 New DoS-resistant OMR

With our snake-eye resistant PKE scheme LWEmongrass, we achieve provable DoS-resistance while nearly matching the performance of PerfOMR [36] (Fig. 1 step (c)).

Constructing this OMR scheme from LWEmongrass follows the same pattern as in [34] (see also Fig. 2). Each recipient generates a LWEmongrass.pk as the clue key, and the sender generates LWEmongrass.Enc(pp,pk,0^{ℓ}) as the clue. Again, the recipient generates $\mathsf{ct}_{\mathsf{sk}}$ encrypting

[11] [36] used the Ring-LWE version of shortLWE, but the main advantages and disadvantages are unaffected, so we illustrate using the one defined in Definition 3.5. Also, [36] has two OMR constructions PerfOMR1 and PerfOMR2, mainly differing in parameter choice and suffering from the same DoS attacks. For better readability, we refer to them as PerfOMR.

Algorithm 2 Practical Snake-eye Resistant OMR Scheme

```
1: procedure $\mathsf{DoS\text{-}PerfOMR.GenParam}(1^\lambda, \epsilon_p, \epsilon_n)$
2:     Choose BFV parameters $\mathsf{pp}_{\mathsf{BFV}}$ such that BFV is secure and correct.
3:     Choose $\ell, t$ such that
         1. $(\frac{4r+1}{t})^\ell \leq \epsilon_p$
         2. $\mathsf{pp}_{\mathsf{LWEmongrass}} = (\dots, r, \dots) \leftarrow \mathsf{LWEmongrass.GenParams}(1^\lambda, (\ell, t, 2, \mathcal{T}, \chi_\sigma, 1))$ where $r$ is
            chosen such that $\mathsf{LWEmongrass.Dec}(\mathsf{pp}, \cdot, \cdot)$ is correct with probability $> 1 - \epsilon_n - \mathsf{negl}(\lambda)$.
         3. $t$ is the plaintext space of the underlying BFV scheme.
4:     return $\mathsf{pp} = (\mathsf{pp}_{\mathsf{BFV}}, \mathsf{pp}_{\mathsf{LWEmongrass}}, \epsilon_p, \epsilon_n)$
5: procedure $\mathsf{DoS\text{-}PerfOMR.KeyGen}(\mathsf{pp})$
6:     $(\mathsf{sk}_{\mathsf{LWEmongrass}}, \mathsf{pk}_{\mathsf{LWEmongrass}}) \leftarrow \mathsf{LWEmongrass.KeyGen}(\mathsf{pp}_{\mathsf{LWEmongrass}})$
7:     $(\mathsf{sk}_{\mathsf{BFV}}, \mathsf{pk}_{\mathsf{BFV}}) \leftarrow \mathsf{BFV.KeyGen}(\mathsf{pp}_{\mathsf{BFV}})$
8:     $\mathsf{ct}_{\mathsf{sk}_{\mathsf{LWEmongrass}}} \leftarrow \mathsf{BFV.Enc}(\mathsf{pk}_{\mathsf{BFV}}, \mathsf{sk}_{\mathsf{LWEmongrass}})$
9:     return $(\mathsf{sk} = (\mathsf{sk}_{\mathsf{BFV}}), \mathsf{pk} = (\mathsf{pk}_{\mathsf{clue}} = \mathsf{pk}_{\mathsf{LWEmongrass}}, \mathsf{pk}_{\mathsf{det}} = (\mathsf{pk}_{\mathsf{BFV}}, \mathsf{ct}_{\mathsf{sk}_{\mathsf{LWEmongrass}}})))$
10: procedure $\mathsf{DoS\text{-}PerfOMR.GenClue}(\mathsf{pp}, \mathsf{pk}_{\mathsf{clue}}, x)$
11:     $\vec{p} \leftarrow (0, \dots, 0) \in \mathbb{Z}_2^\ell$
12:     $c \leftarrow \mathsf{LWEmongrass.Enc}(\mathsf{pp}_{\mathsf{LWEmongrass}}, \mathsf{pk}_{\mathsf{clue}}, \vec{p})$
13:     return $c$
14: procedure $\mathsf{DoS\text{-}PerfOMR.Retrieve}(\mathsf{pp}, \mathsf{BB} = (x_i, c_i)_{i \in [N]}, \mathsf{pk}_{\mathsf{det}}, \bar{\gamma})$
15:     Use BFV to homomorphically decrypt each $c_i$ using $\mathsf{ct}_{\mathsf{sk}_{\mathsf{LWEmongrass}}}$ and $\mathsf{pp}_{\mathsf{LWEmongrass}}$ to obtain
        $N$ encryption bit vectors $\vec{\mathsf{PV}}_i$ each of length $\ell$        ▷ See [36, Alg.1] for a concrete realization
16:     Homomorphically compute $\mathsf{PV}_i \leftarrow \prod_j (1 - \vec{\mathsf{PV}}_i[j])$, so that $\mathsf{PV}_i$ encrypts 1 iff the message is
        encrypted under $\mathsf{sk}_{\mathsf{LWEmongrass}}$ encrypted under $\mathsf{ct}_{\mathsf{sk}_{\mathsf{LWEmongrass}}}$.
17:     Use $\mathsf{PV}_i$ and $x_i$ to generate the digest $M$ via BFV: if $\mathsf{PV}_i$ encrypts 1, $x_i \in \mathsf{Decode}(M, \mathsf{sk})$
        and otherwise, $x_i \notin \mathsf{Decode}(M, \mathsf{sk})$.        ▷ See [36, Alg.2&3] for a concrete realization
18:     return $M$
19: procedure $\mathsf{DoS\text{-}PerfOMR.Decode}(M, \mathsf{sk})$
20:     Decrypt and decode $M$ with $\mathsf{sk}_{\mathsf{BFV}}$.        ▷ See [36, Alg.4] for a concrete realization
```

LWEmongrass.sk under BFV.pk, and the detector uses the homomorphic decryption circuit proposed in [36] to homomorphically decrypt each clue (with $\mathsf{ct}_{\mathsf{sk}}$). The rest proceeds exactly the same as in [34, 36], so we omit the details.

Since LWEmongrass is indeed snake-eye resistant, it follows that this new OMR construction is DoS-resistant.

Efficiency improvement compared to OMRp2. Since LWEmongrass public key size is only $O(\ell \cdot n \log(q))$, the clue key size of our new OMR construction is much smaller than OMRp2 in [34]. Furthermore, the range r for LWEmongrass is also much smaller than r for PVW when β_1, β_2 are small (e.g., $\beta_1 = \beta_2 = 1$ for ternary or binary secrets). Therefore, for the same false positive rate $\epsilon_p = (\frac{4r+1}{q})^\ell$, ℓ can be set to a smaller number compared to [34]. Since the detector needs to perform a homomorphic decryption process for all ℓ bits, a smaller ℓ helps to reduce the computation work of the detector.

Moreover, as introduced in [36], a smaller r leads to a more efficient homomorphic decryption circuit. In [34], the homomorphic decryption circuit is designed to be a linear transformation followed by a degree-$(q-1)$ polynomial $f(x)$ (to check whether the linear transformation result is within $[-r+q/2, r+q/2]$; if so, return 1; otherwise, return 0). However, [36] observes that another polynomial $f'(x) = 1 - \prod_{i \in [r]}(x^2 - r^2)^{q-1}$ has the same functionality as $f(x)$, but only needs $r + \log(q)$ homomorphic multiplications, which is fewer than q multiplications with $r \ll q$; and utilizing this technique, our scheme's homomorphic decryption is much faster than the one in [34].

[36] also proposes some additional optimizations, but since those are irrelevant to the main topic of this paper (e.g., [36] also improved the encoding scheme), we omit the details. Our construction formally described in Algorithm 2 incorporates all the optimizations introduced in [36].

Theorem 7.2. *The scheme* $\mathsf{DoS\text{-}PerfOMR}$ *in Algorithm 2 is a DoS-resistant* OMR *scheme for* $N < D \cdot t/2$*, assuming the correctness, key-privacy, and snake-eye-resistance of* $\mathsf{LWEmongrass}$ *(Theorem 5.2, Theorem 5.3), and the correctness and CPA-security of* BFV *leveled HE.*

Proof sketch. The proof remains very similar to the proof in [34], except that they prove the theorem with a conjecture that PVW is snake-eye resistant, while we prove it under the fact that our scheme $\mathsf{LWEmongrass}$ is indeed snake-eye resistant under standard lattice assumption. Due to the similarity and that the proof is relatively straightforward, we simply sketch the proof.

We first construct an indicator $\mathcal{I}$ with $\epsilon_\mathrm{n}, \epsilon_\mathrm{p}$. The indicator simply performs $\mathsf{LWEmongrass.Dec}$ using sk to each clue c_i. By the correctness of $\mathsf{LWEmongrass}$, indicator completeness holds. Furthermore, by the $((4r+1)/q)^\ell$-snake-eye-resistance property of $\mathsf{LWEmongrass}$, the collision resistance property holds. Namely, if an adversary breaks the collision resistance, there is an adversary that breaks the snake-eye-resistance property of $\mathsf{LWEmongrass}$: the clue that breaks the DoS resistance property can be decrypted into the same message under two honestly generated $\mathsf{LWEmongrass}$ secret keys with $> \epsilon_\mathrm{n} = ((4r+1)/q)^\ell$ probability.

Then, by the correctness of BFV together with line 17, DoS-completeness and DoS-soundness hold (m_i is included in the correctly decoded payload, iff c_i is homomorphically decrypted into 0^ℓ).

Lastly, the computational privacy is trivially implied by the CPA-security of BFV (such that $\mathsf{sk}_{\mathsf{LWEmongrass}}$ remains private) and the key-privacy property of $\mathsf{LWEmongrass}$ (which implies computational privacy given that $\mathsf{sk}_{\mathsf{LWEmongrass}}$ is private). □

Remark 7.3. We briefly discuss the two properties we skipped in Section 7.3: compactness and key-unlinkability. Compactness requires that $|M| \ll |\mathsf{BB}|$. We skip the details because achieving compactness mainly depends on how the compression at line 17 is realized and is irrelevant to the paper. Using the compression technique of prior works, we achieve the same compactness guarantee as [34, 36].

The strongest form of key-unlinkability defined in [34, Def 9.2] is full-key-unlinkability. It states that (1) the recipient can generate multiple clue keys and detection keys with the same secret key and all the properties still hold regardless of which keys are used; (2) all these keys are unlinkable to each other (i.e., indistinguishable from keys generated using independently and honestly generated secret keys). Our scheme achieves such full-key-unlinkability as in [34] under standard Ring-LWE assumption (by the fact that $\mathsf{LWEmongrass}$ and BFV public keys are indistinguishable from random samples and the CPA security of BFV).

	Detector Runtime (ms/msg)	Clue Key Size (KB)	Clue Size (bytes)	Detector Key Size (MB)	Digest Size (bytes/msg)	Recipient Runtime (ms)	DoS-resistance
OMRp2 [34, 35]	1-thread: 59.8 2-thread: 30.1	132.81	956	139	1.08	11	Conjectured in [34], proven by Lemma 4.3
PerfOMR1 [36]	1-thread: 4.1 2-thread: 2.1	2.13	2181	171	2.71	21	No (attacks in Section 5.1)
PerfOMR2 [36]	1-thread: 21.5 2-thread: 11.3	0.56	583	140	1.08	11	
DoS-PerfOMR Section 7.5	1-thread: 4.9 2-thread: 2.6	4.73	1996	183	2.71	20	Yes (Theorem 5.3)

Table 1: Comparison with prior OMR constructions for $N = 2^{19}, \gamma = \bar{\gamma} = 50$.

8 Evaluation

We implemented the above DoS-PerfOMR scheme in an open-source C++ library [23]. Our code extends the OMR library [43] and uses the SEAL [41] and PALISADE [46, 6] libraries. We enabled SEAL's HEXL hardware acceleration [12] using Intel's AVX512-IFMA52 CPU features.

We compare our constructions to the conjectured-DoS-resistant construction OMR-OPT introduced in [34][12] and the non-DoS-resistant constructions PerfOMR1 and PerfOMR2 introduced in [36]. We benchmark our implementation, alongside the prior constructions, on a Google Compute Cloud n4-standard-8 instance (32GB RAM), reporting 1-thread and 2-thread runtime as in [36]. For a fair comparison, we benchmark the prior schemes using the same instance (also with Intel HEXL acceleration), resulting in roughly 2x speedup compared to [34, 35, 36].

Parameters. We choose the number of messages to be $N = 2^{19}$, and let the total number of pertinent messages (and the bound estimated by the recipient) be $\gamma = \bar{\gamma} = 50$; we also set the payload size to 612 bytes (based on Zcash transactions [24]), all same as in [34, 35, 36].

All the computational security estimations below are done using the up-to-date version of [4]. For OMR-OPT, we reuse the parameters in [35]. Then, for the LWEmongrass used in DoS-PerfOMR we choose $n_1 = 936, n_2 = 760, q = 65537, \sigma = 0.5, k = 1$ with ternary secrets for KeyGen and binary secrets for Enc which guarantees computational 128-bit security for the underlying LWE (with ternary secrets) and rhLWE (with binary secrets) to hold[13], to guarantee CPA-security and key-privacy, which are essential. In addition, the above parameter guarantees a computational security of 100-bit and a statistical security of 40-bit for the underlying corLWE (with ternary secrets) to hold, under which our DoS-PerfOMR is proved to be DoS-resistant. Then, we set $r = 79, \ell = 3$ to guarantee an $\epsilon_{\mathrm{n}} < 2^{-30}, \epsilon_{\mathrm{p}} < 2^{-22}$ (equal or smaller than the prior works [34, 35, 36]). For the underlying BFV scheme, we set ring dimension $D = 2^{15}$, plaintext modulus $t = 65537$, and ciphertext modulus $Q \approx 2^{970}$, which guarantees 105-bit security [4].

Performance analysis. As shown in Table 1, our construction is DoS-resistance as OMRp2 in [34, 35] (proven by our result), while having a comparable performance with PerfOMR1 in [36]. Our construction is about 12x faster than OMRp2 in terms of the detector runtime, which is indeed the main cost of the OMR construction, and only about 1.2x slower than PerfOMR1. Furthermore, our clue key size is about 28x smaller than OMRp2 and only about 2.2x larger than PerfOMR1. Note that PerfOMR2 uses a different parameterization that minimizes the clue key size and clue

[12]We benchmark the improved version from [35].

[13]Note when using LWE with $n = 760, q, \sigma$, it gives 131.5 bits of security to cover the $2|\chi| + 1$ security loss from rhLWE.

		$\gamma = \bar{\gamma} = 50$			
	N	Amortized runtime (ms/msg)	Total runtime (s)	Amortized digest size (bytes/msg)	Total digest size (MB)
DoS-PerfOMR	2^{19}	4.95	2597.53	2.71	1.35
	2^{21}		10663.81	0.68	
	2^{23}		42467.92	0.17	

		$N = 2^{19}$			
	$\gamma = \bar{\gamma}$	Amortized runtime (ms/msg)	Total runtime (s)	Amortized digest size (bytes/msg)	Total digest size (MB)
DoS-PerfOMR	50	4.95	2597.53	2.71	1.35
	100	5.44	2850.74	4.87	2.43
	150	5.83	3067.73	7.04	3.52

Table 2: Scalability of our construction DoS-PerfOMR.

size, while sacrificing detector runtime. Thus, it has different trade-offs compared to PerfOMR1 and thereby our construction.

Scalability. As shown in Table 2, our construction runtime essentially grows linearly with N and grows slightly with $\bar{\gamma}$ as well. Digest size remains unchanged with different N but grows linearly with $\bar{\gamma}$. These scalability behaviors are essentially the same as all prior single-server practical OMR constructions [34, 35, 36].

Attacks on [36]. We implement the attack on PerfOMR discussed in Section 5.1 (adapted according to the ternary sparse secret as used in [36]). By maliciously crafting a clue $(\vec{a}, b)$, with $\vec{a}$ having r ones and $b = 0$, we observe that the message attached with this clue is detected as pertinent in all the 10 trials of different honestly generated detection keys. Such attack easily generalizes as discussed in Section 5.1. Furthermore, by adding 100 such maliciously crafted clues, we observe that all the honest recipients show "overflow" during the detection process, using $\bar{\gamma} = 50$ (even without true pertinent messages). Thus, such an attack greatly affects the completeness of retrieval.

9 Relation and Application to Robust Encryption

As briefly discussed in Section 2, in [2], a PKE property called *strong robustness* is introduced. At a high-level, a PKE scheme is strongly robust, if for any PPT adversary, it is hard to find a ciphertext that is *valid* under two honestly generated secret keys. Below we formally recall this definition (we introduce a parameter δ such that $\delta = \mathsf{negl}(\lambda)$ corresponds to the definition in [2]).

Definition 9.1 (δ-strong robustness of PKE)**.** A PKE scheme is δ-strongly robust if the following holds: let $\mathsf{pp} \leftarrow \mathsf{PKE.GenParam}(1^\lambda, \mathsf{aux})$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp}), (\mathsf{pk}_2,\mathsf{sk}_2) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{pp})$; for any PPT adversary $\mathcal{A}$, let $\mathsf{ct} \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{pk}_2)$:

$$\Pr\left[\mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_1, \mathsf{ct}) \neq \perp \wedge \mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_2, \mathsf{ct}) \neq \perp\right] \leq \delta.$$

It is easy to see that δ-strong-robustness implies δ-snake-eye resistance (i.e., any PKE scheme that is δ-strongly robust is δ-snake-eye resistant). However, unfortunately, all existing PKE schemes with robustness that are based on lattices [29, 40] rely on the same paradigm: a combination of a key encapsulation mechanism together with symmetric key encryption. These constructions are unsuitable for our application in OMR since we would need to homomorphically decrypt symmetric key encryptions, which is extremely inefficient (e.g., the state-of-the-art FHE-based AES [53] takes

26.2 seconds to decrypt a single ciphertext, while our OMR detector runtime only takes < 0.01 sec per decryption).

9.1 Snake-eye Resistance Implies Strong Robustness

We show that δ-snake-eye resistance also implies δ-strong-robustness via a black box transformation. At a high level, if Π is a δ-snake-eye resistant PKE scheme, then changing the encryption algorithm by appending an encryption of 0 to the original ciphertext results in a δ-strongly robust PKE scheme. Decryption first checks if the appended ciphertext is 0, and if not, returns $\perp$. Since δ-snake-eye resistance guarantees that it is impossible for a ciphertext to be decrypted to 0 under two honestly generated secret keys, the decryption will return non-$\perp$ messages for any two keys with at most δ probability. We formalize this intuition in Algorithm 3 and Theorem 9.2.

Algorithm 3 Black-box transformation from a snake-eye resistant PKE Π to a strongly robust PKE Π'

```
 1: procedure $\Pi'.\mathsf{GenParam}(1^\lambda, \mathsf{aux})$
 2:     return $\Pi.\mathsf{GenParam}(1^\lambda, \mathsf{aux})$
 3: procedure $\Pi'.\mathsf{KeyGen}(\mathsf{pp})$
 4:     return $\Pi.\mathsf{KeyGen}(\mathsf{pp})$
 5: procedure $\Pi'.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m)$
 6:     $c_1 \leftarrow \Pi.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m)$
 7:     $c_2 \leftarrow \Pi.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, 0)$        $\triangleright$ 0 should be $0^\ell$ if the input space is a vector of $\ell$ elements.
 8:     return $(c_1, c_2)$
 9: procedure $\Pi'.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct} = (c_1, c_2))$
10:     If $\Pi.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, c_2) \neq 0$: return $\Pi.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, c_1)$
11:     Else: return $\perp$
```

Theorem 9.2. *If Π is a δ-snake-eye resistant PKE scheme, then Π' is a δ-strongly robust PKE scheme.*

Proof sketch. Suppose that Π' is not δ-strongly robust, then there exists a PPT adversary that outputs a ciphertext $\mathsf{ct} = (c_1, c_2)$ such that $\Pi'.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_1, \mathsf{ct}) \neq \perp$ and $\Pi'.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_2, \mathsf{ct}) \neq \perp$ where $\mathsf{sk}_1, \mathsf{sk}_2$ are two honestly generated secret keys with probability greater than δ. This means that $\Pi.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_1, c_2) = \Pi.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_2, c_2) = 0$ with probability greater than δ, breaking the δ-snake-eye resistance of Π. $\square$

9.2 Robust LWEmongrass

While the above transformation implies that LWEmongrass can be easily made robust via a black box transformation, there is a direct transformation with better efficiency. Specifically, while the black-box transformation above increases the ciphertext size grow by a factor of 2, our transformation of LWEmongrass only grows the ciphertext slightly.

Recall that in LWEmongrass (or in general in any LWE-based PKE scheme) the ciphertext has the form $\mathsf{ct} = (\vec{a}, \vec{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^\ell$ encrypting a plaintext $\vec{m} \in \mathbb{Z}_p^\ell$. Then, instead of encrypting $\vec{m}$, we let the scheme encrypt $\vec{m}||0^\ell$, making it $\mathsf{ct}' = (\vec{a}', \vec{b}') \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{2\ell}$. This means that the ciphertext size grows from $(n+\ell)\log(q)$ to $(n+2\ell)\log(q)$ (while the black-box transformation has ciphertext size $2(n+\ell)\log(q)$). This is useful when ℓ is small. The resulting scheme is δ-strongly robust. One can also instead append $0^{\ell'}$ and set ℓ' (e.g., to achieve smaller δ) without affecting ℓ. We formalize this intuition with Algorithm 4 and Theorem 9.3.

Algorithm 4 Strongly robust LWEmongrass

```
1: procedure $\mathsf{LWEmongrass}'.\mathsf{GenParam}(1^\lambda, \mathsf{aux} = (\ell, q, p, \dots))$
2:     return $\mathsf{LWEmongrass}.\mathsf{GenParam}(1^\lambda, \mathsf{aux}' = (2\ell, q, p, \dots))$ ▷ $\mathsf{aux}'$ simply changes the $\ell$ in $\mathsf{aux}$ to $2\ell$
3: procedure $\mathsf{LWEmongrass}'.\mathsf{KeyGen}(\mathsf{pp})$
4:     return $\mathsf{LWEmongrass}.\mathsf{KeyGen}(\mathsf{pp})$
5: procedure $\mathsf{LWEmongrass}'.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \vec{m} \in \mathbb{Z}_p^\ell)$
6:     return $\mathsf{LWEmongrass}.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \vec{m}||0^\ell)$
7: procedure $\mathsf{LWEmongrass}'.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \mathsf{ct})$
8:     $\vec{m} \leftarrow \mathsf{LWEmongrass}.\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) \neq 0$
9:     If $\vec{m}[\ell+1, 2\ell] \neq 0^\ell$, return $\perp$.
10:    return $\vec{m}[1, \ell]$
```

Theorem 9.3. *Given that* LWEmongrass *is δ-snake-eye resistant,* LWEmongrass$'$ *is δ strongly robust.*

Proof sketch. The proof is essentially the same as the proof for Theorem 9.2: if one can break the strong robustness of LWEmongrass$'$, one can generate $\mathsf{ct} = (\vec{a}, \vec{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{2\ell}$ that decrypts to non-$\perp$ under two honest generated LWEmongrass secret keys. Therefore, $\mathsf{ct}' := (\vec{a}, \vec{b}[\ell+1, 2\ell])$ is a ciphertext that breaks the snake-eye resistance of LWEmongrass. ☐

Efficiency analysis. As mentioned, the ciphertext size only grows a little from LWEmongrass. However, more importantly, to decrypt for ℓ bits (for simplicity assume $p = 2$), only ℓ homomorphic decryption circuits are needed. Using existing robust lattice-based PKE schemes [29, 40], one needs to decrypt at least 128 bits to obtain a symmetric key first and then perform a symmetric key decryption using this key. This is very inefficient if the decryption needs to be done using FHE like the setting of OMR. Therefore, our construction is more efficient than the existing ones under such a setting (i.e., small ℓ, and a simpler better decryption circuit is preferred).

10 Extension of PKE Snake-eye Resistance

While Definition 4.1 is enough for the application we are interested in, we can prove an even stronger property for PVW and our scheme.

In this section, we consider a generalization: $\mathcal{A}$ not only outputs a ciphertext ct, but also a difference between the two decrypted plaintexts $d \in \mathcal{P}$. The adversary wins as long as the two decrypted results' difference is d. Formally, it is defined as follows.

Definition 10.1 (δ-general-snake-eye resistance of PKE)**.** A PKE scheme is δ-general-snake-eye-resistant if the following holds: let $\mathsf{pp} \leftarrow \mathsf{PKE}.\mathsf{GenParam}(1^\lambda)$, $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{PKE}.\mathsf{KeyGen}(\mathsf{pp})$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow$

$\mathsf{PKE.KeyGen}(\mathsf{pp})$; for any PPT adversary $\mathcal{A}$, let $(\mathsf{ct}, d) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{pk}_2)$, where $d \in \mathcal{P}$, $\mathcal{P}$ being the plaintext space, it holds that:

$$\Pr[\mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_1, \mathsf{ct}) = \mathsf{PKE.Dec}(\mathsf{pp}, \mathsf{sk}_2, \mathsf{ct}) + d] \leq \delta.$$

General-snake-eye-resistance of PVW and $\mathsf{LWEmongrass}$. Now we show that both PVW (Definition 3.3) and our new construction (Algorithm 1) are also snake-eye-resistant with respect to this generalized definition. The proof is almost exactly the same with a small difference on the δ and the range checked in the proof.

Essentially, we are leveraging the fact that both of these schemes are linearly homomorphic (with scalars). In other words, for a ciphertext $\mathsf{ct} = \mathsf{Enc}(m_1)$ of these two schemes, it satisfies that $\mathsf{Dec}(\mathsf{ct}') = m_2 = m_1 + d$ where $\mathsf{ct}' \leftarrow \mathsf{ct} + \Delta \cdot d$ for $\Delta = \lfloor q/t \rfloor$ (recall that these schemes encode the plaintext first during encryption). The only caveat is that if $m_2 = 0, m_1 \neq 0, k \neq 0$, the $\Delta \cdot m_2 = 0 \equiv q \mod q$, but $\Delta \cdot (m_1 + k) \neq q$. Instead, $\Delta \cdot (m_1 + k) \in [q - 2, q]$. To capture this caveat, instead of proving $\delta = (\frac{4r+1}{q})^\ell$ for the weaker snake-eye-resistance property as in Lemma 4.3 and Theorem 5.3, we prove $\delta = (\frac{4r+5}{q})^\ell$ to accommodate this "off-by-two" issue for this general-snake-eye resistance formally as follows.

Lemma 10.2 (PVW is $(\frac{4r+5}{q})^\ell$-general-snake-eye resistant)**.** *The PVW PKE scheme (Definition 3.3) with* $\mathsf{pp} = (n, \ell, w, q, \mathcal{U}, \sigma, \cdot)$*, is* $(\frac{4r+5}{q})^\ell$*-snake-eye resistant (Definition 4.1) if* $\mathsf{corLWE}_{n,\ell,w,q,\mathcal{U}^n,\chi_\sigma}$ *(Definition 4.2) is hard.*

Proof. The proof follows the same idea as for Lemma 4.3. (The difference is marked in blue).

Given a PPT adversary $\mathcal{A}$ that breaks the snake-eye resistance property of PVW, we construct an adversary $\mathcal{A}'$ that breaks Lemma 6.1 as follows.

Given $(A_1, \vec{b}_1, A_2, \vec{b}_2, \mathsf{sk}_1 - \mathsf{sk}_2)$, we directly feed $\mathsf{pk}_1 = (A_1, \vec{b}_1), \mathsf{pk}_2 = (A_2, \vec{b}_2)$ into $\mathcal{A}$. Upon receiving $(\vec{a}, \vec{b})$ and d, if $|\vec{a}(\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j}) - \lfloor \frac{q}{p} \rfloor \cdot d| \leq 2r + 2$ for all $j \in [\ell]$, output "the input is a valid corLWE sample"; otherwise, output "the input is a random sample".

If the original $(A_1, \vec{b}_1), (A_2, \vec{b}_2)$ are indeed valid LWE samples, then, it satisfies that $|\vec{a},\mathsf{sk}_{i,j} - \vec{b}_j - \lfloor \frac{q}{p} \rfloor \cdot m_i| \leq r+$ for $i \in \{1,2\}, j \in [\ell]$ where $m_1 = m_2 + d \in \mathcal{P}$, and thus $|\vec{a}(\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j}) - \lfloor \frac{q}{p} \rfloor \cdot d| \leq 2r+2$ via triangle inequality.

On the other hand, if the original $(A_1, \vec{b}_1), (A_2, \vec{b}_2)$ are random sample, $\mathcal{A}$ receives no information about $\mathsf{sk}_1, \mathsf{sk}_2$ (which are both uniformly randomly sampled). Thus, for $\vec{a} \neq 0^n$, we have $\Pr[\forall j \in [\ell], |\vec{a}(\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j})| \leq 2r + 2] = (\frac{4r+5}{q})^\ell$.

Therefore, for any $\mathcal{A}$ with noticeable advantage δ over $(\frac{4r+5}{q})^\ell$, $\mathcal{A}'$ has the same advantage δ. □

Lemma 10.3. *Under* $\mathsf{corLWE}_{n_1,\ell,n_2,q,\mathcal{D}_\beta^n \| \mathcal{U}^k,\chi}$ *is hard, where* $n_1 = n + k$*, the PKE scheme defined in Algorithm 1 is* $(\frac{4r+5}{q})^\ell$*-general-snake-eye resistance.*

Proof sketch. Similarly change all the checks of $2r$ to $2r + 2$ for the proof of Theorem 5.3 as in the blue part of the proof of Lemma 10.2. Everything else follows exactly the same. □

11 Open Questions

Tighter snake-eye resistance. As mentioned in Remark 4.4, the main difference between what we prove and what is conjectured in [34] is that they conjecture $\delta = ((2r+1)/q)^\ell$ while we prove for

$\delta = ((4r+1)/q)^{\ell}$. Note that $\delta = ((2r+1)/q)^{\ell}$ is indeed the bound if ct is *honestly generated* (i.e., encrypted using the Enc interface with some honestly generated public key). Therefore, it is natural to believe that $\delta = ((2r+1)/q)^{\ell}$ even in the malicious case. While this has limited influence in practice, it is an interesting theoretical question whether we can indeed prove snake-eye resistance with $\delta = ((2r+1)/q)^{\ell}$.

Additional parameterization for our LWE variants. In Section 6, we proved that under some secret distributions (i.e., uniform, binary and ternary), corLWE and rhLWE are at least as hard as regular LWE. While the distributions we proved are enough to support our claims about the snake-eye resistance of $\mathsf{LWEmongrass}$ ($\mathcal{T}$ for corLWE and $\mathcal{B}$ for rhLWE), there are other popular secret distributions, like Gaussian distribution, uniform-t-ary (i.e., uniform over $[-t, t]$), sparse secrets (i.e., with some fixed hamming weight $h \ll n$), and so on. For these different distributions, it seems intuitive that we can obtain a reduction from standard LWE to corLWE and/or rhLWE with some security loss using similar proof techniques (e.g., for rhLWE, it seems to hold for any secret distribution as long as the corresponding LHL holds). We leave it for future work to explore and improve the tightness of the reduction. Another interesting question is whether one can prove or disprove rhLWE with $k = O(\log(n))$ is as hard as standard LWE for binary secrets.

Futher generalization of the snake-eye resistance property. While we have discussed some generalizations (in Section 10), there are other natural definitions that are even stronger. For example, can the adversary output a function f such that $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_1) = f(\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_2))$? Or can the adversary outputs a relation R such that $R(\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_1), \mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_2)) = 1$? While all these are intriguing directions, it seems hard for our scheme to satisfy. In particular, taking the function f version of generalization as an example, the ciphertext satisfies that $\vec{a}(\mathsf{sk}_{1,j} - \mathsf{sk}_{2,j}) \approx m_1 - f(m_1)$ for some unkown m_1; thereby $m_1 - f(m_1)$ may be arbitrary. Thus, we cannot check the result against some range r. Whether such definitions can be satisfied by any constructions is thus left for future works to explore.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, (3), July 2008.

[2] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *TCC 2010*, LNCS. Springer, Heidelberg, Germany, Feb. 9–11, 2010.

[3] C. Abou Haidar, A. Passelègue, and D. Stehlé. Efficient updatable public-key encryption from lattices. In J. Guo and R. Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 342–373, Singapore, 2023. Springer Nature Singapore.

[4] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, pages 169–203, 2015.

[5] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *PKC 2012*, LNCS. Springer, Heidelberg, Germany, May 21–23, 2012.

[6] A. A. Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff,

J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca. Openfhe: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022.

[7] G. Beck, J. Len, I. Miers, and M. Green. Fuzzy message detection. ACM CCS 2021.

[8] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE S&P*, pages 459–474, 2014.

[9] J. Bethencourt, D. X. Song, and B. Waters. New techniques for private stream searching. *ACM Trans. Inf. Syst. Secur.*, 12:16:1–16:32, 2009.

[10] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, 2017.

[11] O. Biçer and C. Tschudin. Oblivious homomorphic encryption. Cryptology ePrint Archive, Paper 2023/1699, 2023.

[12] F. Boemer, S. Kim, G. Seifu, F. D. de Souza, V. Gopal, et al. Intel HEXL (release 1.2.5). `https://github.com/intel/hexl`, Aug. 2022.

[13] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. Crystals - kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367, 2018.

[14] S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra, and H. Wu. Zexe: Enabling decentralized private computation. In *2020 IEEE S&P (SP)*, pages 947–964, 2020.

[15] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *CRYPTO 2012*, LNCS. Springer, Aug. 19–23, 2012.

[16] Z. Brakerski and N. Döttling. Hardness of LWE on general entropic distributions. In *EURO-CRYPT 2020, Part II*, LNCS. Springer, Heidelberg, Germany, May 10–14, 2020.

[17] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *45th ACM STOC*, June 1–4, 2013.

[18] J. H. Cheon, D. Kim, D. Kim, J. Lee, J. Shin, and Y. Song. Lattice-based secure biometric authentication for hamming distance. In *ACISP 21*, LNCS. Springer, Heidelberg, Germany, Dec. 1–3, 2021.

[19] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords, 1998.

[20] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *36th FOCS*. IEEE Computer Society Press, Oct. 23–25, 1995.

[21] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: An anonymous messaging system handling millions of users. In *2015 IEEE S&P*, pages 321–338, 2015.

[22] G. Danezis and C. Diaz. Space-efficient private search with applications to rateless codes. In *FC'07*, page 148–162. Springer, 2007.

[23] DoS-PerfOMR implementation. `https://github.com/ObliviousMessageRetrieval/ObliviousMessageRetrieval/tree/dos_perfomr`, Mar. 2024.

[24] Electric Coin Company. Zcash Rust crates. `https://github.com/zcash/librustzcash`. Commit hash: 99d877e22d58610dc43021b831a28286ef353a89.

[25] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `https://ia.cr/2012/144`.

[26] M. Finiasz and K. Ramchandran. Private stream search at almost the same communication cost as a regular search. In L. R. Knudsen and H. Wu, editors, *Selected Areas in Cryptography*, pages 372–389, Berlin, Heidelberg, 2013. Springer.

[27] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013, Part I*, LNCS. Springer, Heidelberg, Germany, Aug. 18–22, 2013.

[28] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS 2010*. Tsinghua University Press, Jan. 5–7, 2010.

[29] P. Grubbs, V. Maram, and K. G. Paterson. Anonymous, robust post-quantum public key encryption. In *EUROCRYPT 2022, Part III*, LNCS. Springer, Heidelberg, Germany, May 30 – June 3, 2022.

[30] S. Jakkamsetti, Z. Liu, and V. Madathil. Scalable private signaling. Cryptology ePrint Archive, Paper 2023/572, 2023.

[31] Y. Jia, V. Madathil, and A. Kate. Homerun: High-efficiency oblivious message retrieval, unrestricted. Cryptology ePrint Archive, Paper 2024/188, 2024.

[32] D. Kim, D. Lee, J. Seo, and Y. Song. Toward practical lattice-based proof of knowledge from hint-mlwe. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 549–580, Cham, 2023. Springer Nature Switzerland.

[33] J. Lee, D. Kim, D. Kim, Y. Song, J. Shin, and J. H. Cheon. Instant privacy-preserving biometric authentication for hamming distance. *IACR Cryptol. ePrint Arch.*, 2018:1214, 2018.

[34] Z. Liu and E. Tromer. Oblivious message retrieval. In Y. Dodis and T. Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 753–783, Cham, 2022. Springer Nature Switzerland. Full version: ePrint Archive 2021; internal citations follow the latter's numbering.

[35] Z. Liu, E. Tromer, and Y. Wang. Group oblivious message retrieval. *S&P 2024*, 2024.

[36] Z. Liu, E. Tromer, and Y. Wang. Perfomr: Oblivious message retrieval with reduced communication and computation. Usenix Security, 2024.

[37] J. Lund. Technology preview: Sealed sender for signal. `https://signal.org/blog/sealed-sender/`, Oct. 2018.

[38] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 2013.

[39] V. Madathil, A. Scafuro, I. A. Seres, O. Shlomovits, and D. Varlakov. Private signaling. USENIX Security 2022, 2022.

[40] V. Maram and K. Xagawa. Post-quantum anonymity of kyber. Cryptology ePrint Archive, Report 2022/1696, 2022. `https://eprint.iacr.org/2022/1696`.

[41] Microsoft SEAL. `https://github.com/Microsoft/SEAL`, Nov. 2020. Microsoft Research, Redmond, WA.

[42] S. Noether. Ring signature confidential transactions for monero. *IACR Cryptology ePrint Archive*, 2015:1098, 2015.

[43] Oblivious message retrieval implementation. `https://github.com/ZeyuThomasLiu/ObliviousMessageRetrieval`, Dec. 2021.

[44] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *CRYPTO 2011*, LNCS. Springer, Heidelberg, Germany, Aug. 14–18, 2011.

[45] R. Ostrovsky and W. E. Skeith. Private searching on streaming data. In *CRYPTO*, 2005.

[46] PALISADE lattice cryptography library (release 11.2). `https://palisade-crypto.org/`, June 2021.

[47] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO 2008*.

[48] S. Pu, S. A. Thyagarajan, N. Döttling, and L. Hanzlik. Post quantum fuzzy stealth signatures and applications. *CCS*, 2023.

[49] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, Sept. 2009.

[50] K. Sako. An auction protocol which hides bids of losers. In *PKC 2000*, LNCS. Springer, Heidelberg, Germany, Jan. 18–20, 2000.

[51] I. A. Seres, B. Pejó, and P. Burcsi. The effect of false positives: Why fuzzy message detection leads to fuzzy privacy guarantees? In I. Eyal and J. Garay, editors, *Financial Cryptography and Data Security*, pages 123–148, Cham, 2022. Springer International Publishing.

[52] J. Szefer. Survey of microarchitectural side and covert channels, attacks, and defenses. *Journal of Hardware and Systems Security*, Sept. 2019.

[53] R. Wang, Y. Wen, Z. Li, X. Lu, B. Wei, K. Liu, and K. Wang. Circuit bootstrapping: Faster and smaller. Cryptology ePrint Archive, Paper 2024/323, 2024. Eurocrypt 2024.

[54] D. I. Wolinsky, H. Corrigan-Gibbs, B. Ford, and A. Johnson. Dissent in numbers: Making strong anonymity scale. In *OSDI 12*, pages 179–182. USENIX, Oct. 2012.