Towards Compact Identity-based Encryption on Ideal Lattices

Huiwen Jia[1,2], Yupu Hu[3], Chunming Tang[1,2] and Lin Wang[4]

[1] School of Mathematics and Information Science, Key Laboratory of Information Security, Guangzhou University, Guangzhou, China
ctang@gzhu.edu.cn

[2] Guangzhou Center for Applied Mathematics, Guangzhou University, Guangzhou, China

[3] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, Shaanxi, China

[4] Science and Technology on Communication Security Laboratory, Chengdu, 610041, China

Abstract. Basic encryption and signature on lattices have comparable efficiency to their classical counterparts in terms of speed and key size. However, Identity-based Encryption (IBE) on lattices is much less efficient in terms of compactness, even when instantiated on ideal lattices and in the Random Oracle Model (ROM). This is because the underlying preimage sampling algorithm used to extract the users' secret keys requires huge public parameters. In this work, we specify a compact IBE instantiation for practical use by introducing various optimizations. Specifically, we first propose a modified gadget to make it more suitable for the instantiation of practical IBE. Then, by incorporating our gadget and the non-spherical Gaussian technique, we provide an efficient preimage sampling algorithm, based on which, we give a specification of a compact IBE on ideal lattice. Finally, two parameter sets and a proof-of-concept implementation are presented. Given the importance of the preimage sampling algorithm in lattice-based cryptography, we believe that our technique can also be applied to the practical instantiation of other advanced cryptographic schemes.

1 Introduction

Identity-based encryption Identity-based encryption (IBE), introduced by Shamir in [Sha84], is considered as a viable alternative to the classical public key encryption, which requires a dedicated infrastructure. Indeed, an IBE scheme avoids a certificate repository by deriving a user's public key from its identity, and the associated private key is extracted by a trusted authority using a master secret key. This simplifies the key generation and distribution in a multi-user system and is particularly attractive in resource constrained environments. The first IBE schemes, based on bilinear maps and on quadratic residue assumptions respectively, appeared in [BF01, Coc01], followed by improvements from various perspectives [JR13, Lew12, Wat09, DG17, BWY11, BGK08]. However, these traditional constructions are vulnerable to quantum attacks due to Shor's algorithm [Sho99].

Lattice-based cryptography Lattice-based cryptography is seen as a desirable alternative to the traditional number theoretic cryptography, due to its presumed security against quantum computers, algorithmic simplicity, and versatility for constructing various advanced schemes. For the basic encryption and signature, lattice-based constructions are the most practically efficient among post-quantum cryptosystems. In July 2022, NIST announced the first four post-quantum algorithms to be standardized, and three of them are lattice-based: Kyber [SAB+20] for public key encryption/KEMs; Dilithium [LDK+22] and Falcon [PFH+22] for digital signatures. These algorithms have an efficiency comparable to their classical counterparts.

When it comes to lattice-based IBE, however, this is far from the case. Even when instantiated on ideal lattice and in the Random Oracle Model (ROM), lattice-based IBE schemes still suffer from inefficiencies, particularly in terms of key size. The reason for this is the low efficiency of the associated *preimage sampling algorithm*, which essentially forms the backbone of the user key extraction procedure in lattice-based IBE schemes. In fact, the preimage sampling algorithm plays a central role in a large fraction of the advanced lattice-based cryptosystems.

Preimage sampling At the heart of many lattice-based schemes is what is known as Ajtai's function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod Q$, where $\mathbf{A} \in \mathbb{Z}_Q^{n \times m}$ is a short and fat random matrix. Ajtai's function actually defines the inhomogeneous short integer solution (ISIS) problem, which is believed to be hard [Ajt96, MR04, GPV08] for appropriate parameters. Given a lattice trapdoor for $\mathbf{A}$, one can efficiently compute a short preimage. However, some early proposals [GGH97, HHP+03] based on lattice trapdoor were broken by statistical attacks [NR06, DN12, YD18], since the preimages leak information from the trapdoor.

Towards the proper use of lattice trapdoors, the preimage sampling algorithm was first formalized by Gentry, Peikert and Vaikuntanathan [GPV08], which samples preimages from a given lattice coset with a specific Gaussian distribution. Since then, it has become an essential building block in most advanced cryptographic applications. From an implementation perspective, however, the algorithm itself is inherently sequential and inefficient. In 2010, Peikert [Pei10] proposed the convolution technique and made the sampling procedure parallelizable, at the cost of a moderate increase in the Gaussian parameter of the preimages, which yields some security loss. In the past decade, preimage sampling has been further improved by a batch of follow-up works [MP12, DP16, Pre15, CGM19, DGPY20, EFG+22, ETWY22, YJW23], with the emphasis on the practical instantiations of the hash-and-sign signatures [GPV08], the simplest application of the preimage sampling algorithm. Basically, these instantiations can be classified into two families: *NTRU trapdoor based* and *gadget based*.

1. NTRU trapdoor based. In 2014, Ducas, Lybashevsky and Prest [DLP14] presented the first practical instantiation over NTRU lattices of the sampler in [GPV08], by exploiting a nearly optimal NTRU trapdoor. This scheme was further developed as Falcon [PFH+22] by integrating the fast Fourier sampler [DP16]. Falcon offers good performance in terms of time and space, but its signing and key generation are rather complex. Espitau et al. proposed a simplified variant of Falcon, called Mitaka [EFG+22], which uses the hybrid sampler [Pre15] for easier implementation at the cost of a moderate security loss. Recently, Espitau et al. [ETWY22] have further optimized Falcon and Mitaka by sampling the preimage from an ellipsoidal discrete Gaussian distribution.

2. Gadget based. The gadget based preimage sampling was invented by Micciancio and Peikert in [MP12]. Following the idea of [Pei10], the sampling procedure of the Micciancio-Peikert framework is decomposed into offline and online phases. The online sampling boils down to the sampling over the lattice $\Lambda_Q^{\perp}(\mathbf{g}) = \{\mathbf{z} \mid \langle \mathbf{g}, \mathbf{z} \rangle = 0 \mod Q\}$ where $\mathbf{g} = (1, b, \cdots, b^{k-1})$ is called a *gadget* vector. As shown in [MP12], sampling over the gadget lattice $\Lambda_Q^{\perp}(\mathbf{g})$ is easy and fast, and the key generation is quiet simple, which offers significant advantages in terms of implementation. In addition, the gadget based framework turns out to be extremely versatile for the construction of advanced primitives [GVW13, GVW15, BVWW16]. However, the gadget based constructions suffer from rather large key sizes. To improve the practicality, Chen, Genise and Mukherjee introduced the notion of approximate trapdoor [CGM19] and proposed to use truncated gadget $\mathbf{f} = (b^l, \cdots, b^{k-1})$ for trapdoor construction. While the improvement is substantial, the size of the gadget-based scheme is still much larger than desired. Recently, Yu, Jia and Wang developed a compact gadget framework in which the gadget used is a square matrix, instead of the short and fat one used in [MP12, CGM19]. This further reduces the key size.

Lattice-based IBE. The first lattice-based IBE scheme was proposed in [GPV08] in the ROM under the LWE and SIS assumptions (GPV-IBE), by using the preimage sampling algorithm devised therein. Subsequently, considerable research related to lattice-based IBE has been conducted from different perspectives, such as weakening the assumptions by removing the random oracle [ABB10a, AFL16, Yam16, KY16], and additional security properties [ABB10b, CHKP10, BLSV18]. These constructions demonstrate, on the theoretical side, the versatility of the preimage sampling algorithm for the construction of lattice-based IBE.

On the practical side, the first (proof-of-concept) implementation of IBE with practical parameters was instantiated on the NTRU lattice [DLP14], and its performance was later improved by a number of software optimizations in [MSO17]. As for implementations on ideal lattices, in 2018, Bert et al. [BFRLS18] mixed the IBE scheme [ABB10a] in the standard model on the Ring-SIS/LWE assumptions with the efficient trapdoor of Peikert and Micciancio [MP12] and provided an efficient implementation. Later, Bert et al. [BEP+21] implemented preimage sampling algorithms on module lattices, relying on the works of [MP12, GM18], and several instantiations on module lattice based schemes were presented, including the IBE in the standard model [ABB10a]. The above two implementations of IBE schemes instantiated on ideal lattice are mainly aimed at demonstrating the time efficiency of the preimage sampling, ignoring the huge key size. For example, if we choose a parameter set in [BFRLS18], the master public key is more than 325 KB for 41-bit security in the classical core-SVP model [ADPS16].

Challenges for compact lattice-based IBE in practice Currently, the state of the art in terms of efficiency is still the GPV-IBE instantiated on structured lattices. Recall that in the GPV-IBE, the fat matrix $\mathbf{A} \in \mathbb{Z}_Q^{n\times m}$ and its associated trapdoor $\mathbf{T}$ represent the master public key and the master secret key respectively. Given any identity $id \in \{0,1\}^*$, the trusted authority extracts a sk_{id} for user id by using $\mathbf{T}$. Specifically, id is first hashed to some $\mathbf{u} \in \mathbb{Z}_Q^n$, then a short vector $\mathbf{x}$ following the discrete Gaussian distribution is output as the corresponding sk_{id} by invoking a preimage sampling algorithm, i.e., $\mathbf{x} \sim D_{\mathbb{Z}^m,\sigma}$ conditioned on $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod Q$. On input a bit $\mu \in \{0,1\}$, the encryption algorithm uses the Dual-Regev scheme [GPV08], which first samples $\mathbf{s} \leftarrow \chi_s^n, \mathbf{e} \leftarrow \chi_e^m, e \leftarrow \chi_e$ from some distributions χ_s, χ_e, then computes $\mathbf{c} = \mathbf{s}^t \cdot \mathbf{A} + \mathbf{e}^t \bmod Q$ and $c = \mathbf{s}^t \cdot \mathbf{u} + e + \lfloor \frac{Q}{2} \rceil \cdot \mu \bmod Q$, and finally outputs $ct = (\mathbf{c}, c)$ as the ciphertext. Using its secret key $\mathbf{x}$, the decryption algorithm computes $z = c - \mathbf{c} \cdot \mathbf{x} = e - \mathbf{e}^t \cdot \mathbf{x} + \lfloor \frac{Q}{2} \rceil \cdot \mu \bmod Q$ and outputs 0 if z is closer to 0 than to $\lfloor \frac{Q}{2} \rceil$; otherwise it outputs 1.

Note that the correctness requires that the absolute value of the error term, dominated by $\mathbf{e}^t \cdot \mathbf{x}$, is less than $\lfloor \frac{Q}{4} \rceil$. Typically, the distribution χ_e is the centered binomial distribution with an appropriate parameter, say η, to hide the plaintext μ under the LWE assumption. Roughly, according to the central limit theorem, $\mathbf{e}^t \cdot \mathbf{x}$ follows a distribution that is very close to a discrete Gaussian distribution with a standard deviation of $\sigma\sqrt{m} \cdot \sqrt{\frac{\eta}{2}}$ and an expectation of 0. Consequently, the decryption failure rate for a single-bit encryption can be approximated by the Gaussian error function as $\delta \approx 1 - \mathsf{erf}\left(\frac{\lfloor Q/4 \rceil}{\sigma\sqrt{m}\cdot\sqrt{\eta}}\right)$. Note that $\sigma\sqrt{m}$ is the expected ℓ_2-norm of the preimage $\mathbf{x}$. This implies that for a reasonable decryption failure rate, the ratio $\frac{Q}{\sigma\sqrt{m}}$ should be greater than $c \cdot 4\sqrt{\eta}$ for some constant c. For instance, having $c = 3$ (i.e. the ratio $\frac{Q}{\sigma\sqrt{m}} = 12 \cdot \sqrt{\eta}$) leads to a decryption failure rate of about $1 - \mathsf{erf}(3) \approx 2^{-15.5}$, which can be reduced small enough by Error Correction Codes (ECC). For current preimage samplers, however, the best achievable ratio $\frac{Q}{\sigma\sqrt{m}}$ is far less than desired. As an illustration, recall that Falcon [PFH+22] uses the GPV sampler [GPV08] and is instantiated on the most compact NTRU lattice, which means that the the standard deviation σ and the lattice dimension m are simultaneously the smallest. Even in this case, the ratio is $\frac{Q}{\sigma\sqrt{m}} \approx 2.3$, which cannot guarantee the correctness of the decryption algorithm when used directly in IBE. This explains why the GPV-IBE based on the NTRU lattice uses very large parameters [MSO17]. The situation becomes even worse when the GPV-IBE is instantiated on ideal lattices.

Our contributions In this work, we present an efficient instantiation of the GPV-IBE [GPV08] on ideal lattice, called SRNSG, by incorporating various optimizations, with the emphasis on compactness. More specifically, the contributions of this paper are summarized as follows.

1. Improved preimage sampling for IBE. As for preimage sampling on ideal lattices, the state of the art is the compact gadget in [YJW23] *combined with* the non-spherical Gaussian in [JHT22], which may offer the most efficient hash-and-sign signature on ideal lattice. According to our analysis, however, it is not the best choice for IBE to balance the security and key size. This is because for the former, we only need the preimage norm to be approximately equal to the modulus Q to ensure the forgery security; while for the latter, we need the preimage norm to be much smaller than Q to reduce the decryption failure rate. To remedy this issue, we present an adapted compact lattice gadget based on [YJW23] to make it more suitable for the instantiation of practical IBE, which can be seen as a combination of [CGM19] and [YJW23]. Besides, we incorporate the non-spherical Gaussian [JHT22] into our improved preimage sampler to reduce the size of the users' secret keys.

2. Practical instantiation of GPV-IBE. By plugging our improved preimage sampling algorithm into the GPV-IBE [GPV08], we obtain a compact IBE based on ideal lattice, named SRNSG. Like LAC [LLZ+18], a candidate in the NIST proposals of round 2, to save the bandwidth as much as possible, we use a smaller modulus Q, together with the BCH code in the decryption algorithm to address the problem of increasing the decryption failure rate caused by using a smaller Q.

3. Proof-of-concept implementation. Finally, we provide new parameter sets and give a proof-of-concept implementation of SRNSG to demonstrate its efficiency. The performance is summarized in Table 1.

Table 1. Summarized performance of SRNSG

Security level	NIST-1	NIST-5
mpk size (in bytes)	4896	10272
ct size (in bytes)	8510	16638
Extract (in cycles)	4,364,517	9,999,207
Enc (in cycles)	1,029,074	2,329,433
Security C / Q	133 / 121	294 / 267

Rode map We begin with preliminary materials in Section 2, followed by our improved preimage sampling in Section 3. Then we present the instantiated GPV-IBE by using our new sampler in Section 4. In Section 5, the concrete security analysis are presented. Besides, we provide new parameter sets and the performance of the implementation. Finally, we draw a conclusion in Section 6.

2 Preliminary

Let $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{N}$ denote the set of real numbers, integers and natural numbers respectively. For positive integer q, let $\mathbb{Z}_q = \{-\lfloor q/2 \rfloor, -\lfloor q/2 \rfloor + 1, \cdots, q - \lfloor q/2 \rfloor - 1\}$ denotes the quotient ring $\mathbb{Z}/(q\mathbb{Z})$. For $a \in \mathbb{Z}$, let $(a \bmod q)$ be the unique integer $a' \in \mathbb{Z}_q$ such that $a = a' \bmod q$. For a real-valued function f and a countable set S, we write $f(S) = \sum_{x \in S} f(x)$ assuming this sum is absolutely convergent.

2.1 Linear algebra and lattices

A vector is denoted by a bold lower case letter, e.g. $\mathbf{x} = (x_1, \ldots, x_n)$, and in column form. The concatenation of $\mathbf{x}_1, \mathbf{x}_2$ is denoted by $(\mathbf{x}_1, \mathbf{x}_2)$. Let $\langle \mathbf{x}, \mathbf{y} \rangle$ be the inner product of $\mathbf{x}, \mathbf{y} \in \mathbb{R}$ and $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ be the ℓ_2 norm of $\mathbf{x}$. A matrix is denoted by a bold upper case letter, e.g. $\mathbf{A} = [\mathbf{a}_1 \mid \cdots \mid \mathbf{a}_n]$, where $\mathbf{a}_i$ denotes the i^{th} column of $\mathbf{A}$. Let $\widetilde{\mathbf{A}} = [\widetilde{\mathbf{a}_1} \mid \cdots \mid \widetilde{\mathbf{a}_n}]$ denote the Gram-Schmidt orthogonalization of $\mathbf{A}$. Let $\otimes$ denote the tensor product. Let $\mathbf{A} \oplus \mathbf{B}$ denote the block diagonal concatenation of $\mathbf{A}$ and $\mathbf{B}$. The largest singular value of $\mathbf{A}$ is denoted by $s_1(\mathbf{A}) = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$. Let $\mathbf{A}^t$ be the transpose of $\mathbf{A}$.

We write $\Sigma \succ 0$, when a symmetric matrix $\Sigma \in \mathbb{R}^{m \times m}$ is positive definite, i.e. $\mathbf{x}^t \Sigma \mathbf{x} > 0$ for all nonzero $\mathbf{x} \in \mathbb{R}^m$. We write $\Sigma_1 \succ \Sigma_2$ if $\Sigma_1 - \Sigma_2 \succ 0$. For any scalar s, we write $\Sigma \succ s$ if $\Sigma - s \cdot \mathbf{I} \succ 0$. If $\Sigma = \mathbf{B}\mathbf{B}^t$, we call $\mathbf{B}$ a square root of Σ. We use $\sqrt{\Sigma}$ to denote any square root of Σ when the context permits it.

Given $\mathbf{B} = [\mathbf{b}_1 \mid \cdots \mid \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ with all $\mathbf{b}_i$'s linearly independent, the lattice generated by $\mathbf{B}$ is $\Lambda(\mathbf{B}) = \{\mathbf{B}\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^n\}$. The dimension of $\Lambda(\mathbf{B})$ is n and $\mathbf{B}$ is called a basis. Let $\Lambda^* = \{\mathbf{y} \in \mathrm{span}(\Lambda) \mid \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}, \forall \mathbf{x} \in \Lambda\}$ be the dual lattice of Λ.

In lattice-based cryptography, the q-ary lattice is of special interest and defined by some $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ as:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \mod q\}.$$

The dimension of $\Lambda_q^{\perp}(\mathbf{A})$ is m and $(q \cdot \mathbb{Z})^m \subseteq \Lambda_q^{\perp} \subseteq \mathbb{Z}^m$. For any $\mathbf{u} \in \mathbb{Z}_q^n$ and $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, the "shifted lattice" is the set

$$\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{z} = \mathbf{u} \mod q\} = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{x}.$$

2.2 Gaussians

The Gaussian function $\rho : \mathbb{R}^m \to (0, 1]$ is defined as $\rho(\mathbf{x}) = \exp(-\pi \cdot \langle \mathbf{x}, \mathbf{x} \rangle)$. Applying a linear transformation given by an invertible matrix $\mathbf{B}$ yields

$$\rho_{\mathbf{B}}(\mathbf{x}) = \rho(\mathbf{B}^{-1}\mathbf{x}) = \exp(-\pi \cdot \mathbf{x}^t \Sigma^{-1} \mathbf{x}),$$

where $\Sigma = \mathbf{B}\mathbf{B}^t$. For any $\mathbf{c} \in \mathrm{span}(\mathbf{B})$, the shifted $\rho_{\sqrt{\Sigma}}$ with center $\mathbf{c}$ is defined as $\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x}) = \rho_{\sqrt{\Sigma}}(\mathbf{x} - \mathbf{c})$. Normalizing $\rho_{\sqrt{\Sigma},\mathbf{c}}$, we obtain the continuous Gaussian distribution $D_{\sqrt{\Sigma},\mathbf{c}}$. Restricting the support of the distribution to the lattice Λ, we get the discrete Gaussian distribution $D_{\Lambda,\sqrt{\Sigma},\mathbf{c}}$. Formally, for any $\mathbf{x} \in \Lambda$,

$$D_{\Lambda,\sqrt{\Sigma},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x})}{\rho_{\sqrt{\Sigma},\mathbf{c}}(\Lambda)}.$$

Let $\eta_\epsilon(\Lambda) = \min\{s > 0 \mid \rho(s \cdot \Lambda^*) \leq 1 + \epsilon\}$ be the smoothing parameter with respect to a lattice Λ and $\epsilon \in (0, 1)$. We write $\sqrt{\Sigma} \geq \eta_\epsilon(\Lambda)$, if $\rho_{\sqrt{\Sigma^{-1}}}(\Lambda^*) \leq 1+\epsilon$. We also use $\eta'_\epsilon(\Lambda) = \eta_\epsilon(\Lambda)/\sqrt{2\pi}$ to denote the scaled smoothing parameter.

Let $D^+_{\mathbb{Z},r}$ be the half integer Gaussian defined by $\rho_r(x)/\rho_r(\mathbb{N})$ for any $x \in \mathbb{N}$. We denote $\mathcal{N}_k(\mathbf{c}, \mathbf{\Sigma})$ as the k-dimensional normal distribution with center $\mathbf{c}$ and covariance $\mathbf{\Sigma}$. If $\mathbf{c} = \mathbf{0}$ and $\mathbf{\Sigma} = \mathbf{I}$, we write $\mathcal{N}_k(\mathbf{c}, \mathbf{\Sigma})$ as $\mathcal{N}_k$.

Lemma 1 ([GPV08]). *Let Λ be an m-dimensional lattice with a basis $\mathbf{B}$, then $\eta_\epsilon(\Lambda) \leq \max_i \|\widetilde{\mathbf{b}}_i\| \cdot \sqrt{\log(2m(1 + 1/\epsilon))/\pi}$, where $\widetilde{\mathbf{b}}_i$ is the i-th vector of $\widetilde{\mathbf{B}}$.*

Lemma 2 ([MR04]). *Let Λ be a lattice, $\mathbf{c} \in \mathrm{span}(\Lambda)$. Then for any $\epsilon \in (0, \frac{1}{2})$ and $s \geq \eta_\epsilon(\Lambda)$, $\rho_s(\Lambda + \mathbf{c}) \in [\frac{1-\epsilon}{1+\epsilon}, 1]\rho_s(\Lambda)$.*

Theorem 1 ([GMPW20]). *For any $\epsilon \in [0, 1)$ defining $\bar{\epsilon} = 2\epsilon/(1-\epsilon)$, a matrix $\mathbf{S}$ of full column rank, a lattice coset $A = \Lambda + \mathbf{a} \subset \mathrm{span}(\mathbf{S})$, and a matrix $\mathbf{T}$ such that $\ker(\mathbf{T})$ is a Λ-subspace and $\eta_\epsilon(\Lambda \cap \ker(\mathbf{T})) \leq \mathbf{S}$, we have*

$$\mathbf{T} \cdot D_{A,\mathbf{S}} \approx_{\bar{\epsilon}} D_{\mathbf{T}A,\mathbf{TS}}.$$

2.3 Cyclotomics

Let $\mathbb{Z}^*_m$ be the set of the $d = \varphi(m)$ integers invertible modulo m. Let ζ_m be the m-th primitive root of 1 and $\Phi_m(X) = \sum_{i \in \mathbb{Z}^*_m}(X - \zeta^i_m) \in \mathbb{Z}[X]$ be the m-th cyclotomic polynomial. We denote by $\mathcal{R} = \mathbb{Z}[\zeta_m] \simeq \mathbb{Z}[X]/(\Phi_m(X))$ the cyclotomic ring of conductor m and by $\mathcal{K} = \mathbb{Q}[\zeta_m] \simeq \mathbb{Q}[X]/(\Phi_m(X))$ the corresponding cyclotomic field. Let $\mathcal{K}_\mathbb{R} = \mathcal{K} \otimes \mathbb{R} = \mathbb{R}[\zeta_m] \simeq \mathbb{R}[X]/(\Phi_m(X))$. In this paper, we focus on the case of power-of-two conductor, i.e. $m = 2^t$ for some $t \in \mathbb{Z}$.

Any $f \in \mathcal{K}$ can be uniquely written as $f = \sum_{i=0}^{d-1} f_i \zeta^i_m$ with $f_i \in \mathbb{Q}$. We call $(f)_c = (f_0, \cdots, f_{d-1}) \in \mathbb{Q}^d$ the coefficient embedding (or coefficient vector) of f. The element of $f \in \mathcal{K}$ can also be identified with the matrix form $\mathcal{M}(f) = [(f)_c \mid (\zeta_m f)_c \mid (\zeta^{d-1}_m f)_c] \in \mathbb{Q}^{d \times d}$.

There are d embeddings of $\mathcal{K}$ fixing over $\mathbb{Q}$. Concretely, for $i \in \mathbb{Z}^*_m$, the embedding σ_i is defined by $\sigma_i(\zeta_m) = \zeta^i_m$. These d embeddings are the singular values of $\mathcal{M}(f)$. The canonical embedding of $f \in \mathcal{K}$ is $\sigma(f) = (\sigma_i(f))_{i \in \mathbb{Z}^*_m} \in \mathcal{K}^d$. The conjugate of f is denoted by f^*.

2.4 Identity-based encryption

- $\mathsf{Setup}(1^\lambda)$: on input the security parameter 1^λ output the master public key and master secret key pair (mpk, msk).
- $\mathsf{Extract}(msk, id)$: given msk and a user identity $id \in \{0,1\}^*$, output a secret key sk_{id} for that identity.
- $\mathsf{Enc}(mpk, id, \mu)$: on input mpk, id and a plaintext μ, output a ciphertext ct.
- $\mathsf{Dec}(sk_{id}, ct)$: given sk_{id} and ct, output a plaintext μ.

The correctness condition is that for all identities id, Dec correctly decrypts a ciphertext encrypted to id, given the sk_{id} produced by $\mathsf{Extract}$.

3 Our Improved Preimage Sampler

In this section, we present our improved preimage sampler by incorporating a new gadget and the non-spherical Gaussian.

3.1 A new gadget sampler more suitable for IBE

In this subsection we give a new gadget that is more suitable for practicalisation of advanced cryptographic schemes. Notice that compared with [MP12, CGM19], the most compact gadget is proposed by Yu, Jia and Wang [YJW23] that uses a square matrix as the gadget, instead of a fat one. However, by our analysis, it is not the best choice for IBE on ideal lattice, since the extreme compactness increases the decryption failure rate when used in IBE. We present a modified gadget that may be seen as a combination of [CGM19] and [YJW23]. We begin with the description of our gadget, followed by proving the simulatability of the gadget sampler, which is a crucial property in the security proof.

Our gadget works with a composite modulus $Q = pq$ where p, q are positive integers, as in [YJW23]. Instead of using the square matrix $p \cdot \mathbf{I}$ as the gadget, we choose the gadget as in [MP12, CGM19], but the semi-random sampling technique [YJW23] is retained. In more detail, let b be a small integer, $w = \lceil \log_b q \rceil$ and let $\mathbf{g} = (1, b, \ldots, b^{w-1}) \in \mathbb{Z}^w$. Given a target $t \in \mathbb{Z}_Q$, our sampler outputs some $\mathbf{z} = (z_0, \ldots, z_{w-1}) \in \mathbb{Z}^w$ following discrete Gaussian such that $\langle \mathbf{f}, \mathbf{z} \rangle = t - e \mod Q$ for some small e, where $\mathbf{f} = p \cdot \mathbf{g}$ is the gadget vector in our sampler. We note the bijection $\tau : \mathbb{Z}_Q \mapsto \mathbb{Z}_p \times \mathbb{Z}_q$ defined by $\tau(t) = (t_p, t_q)$ such that $t = pt_q + t_p$. The main idea of our algorithm is to deterministically treat the remainder t_p as the approximation error e as in [YJW23], then sample $\mathbf{z}$ over the coset $\Lambda^{\perp}_{t_q}(\mathbf{g}^t)$ as in [MP12, CGM19]. A formal description is given in Algorithm 1.

It is straightforward to define $\mathsf{ApproxGadget}(\mathbf{t}, r, p, q)$ for $\mathbf{t} \in \mathbb{Z}_Q^n$ by independently calling Algorithm 1 on each entry of $\mathbf{t}$. The correctness of Algorithm 1 is shown in Lemma 3.

Lemma 3. *Algorithm 1 is correct. More precisely, let $p, q > 0$ be integers, $Q = pq$, $r > 0$ and $t \in \mathbb{Z}_Q$ such that $\tau(t) = (t_p, t_q)$. Then* $\mathsf{ApproxGadget}(t, r, p, q)$ *outputs $\mathbf{z}$ such that $\mathbf{z} \sim D_{\Lambda^{\perp}_{t_q}(\mathbf{g}^t), r}$ and $\langle \mathbf{f}, \mathbf{z} \rangle = t - t_p \bmod Q$.*

Algorithm 1: $\mathsf{ApproxGadget}(t, r, p, q)$

Require: a target $t \in \mathbb{Z}_Q$, a positive real $r > 0$ and integers $p, q > 0$ with $Q = pq$

Ensure: a vector $\mathbf{z} \sim D_{\mathbb{Z}^w, r}$ conditioned on $\langle \mathbf{f}, \mathbf{z} \rangle = t - e \bmod Q$ for some $e \in \mathbb{Z}_p$.

1: $(t_p, t_q) \leftarrow \tau(t)$

2: sample $\mathbf{z} \leftarrow D_{\Lambda_{t_q}^{\perp}(\mathbf{g}^t), r}$

3: **return** $\mathbf{z}$

Proof. In Algorithm 1, $\mathbf{z}$ is sampled from $D_{\Lambda_{t_q}^{\perp}(\mathbf{g}^t), r}$, with q being the modulus, hence $\langle \mathbf{g}, \mathbf{z} \rangle = t_q \bmod q$, that is, $p(\langle \mathbf{g}, \mathbf{z} \rangle - t_q) = 0 \bmod Q$. Immediately, we have $\langle \mathbf{f}, \mathbf{z} \rangle = t - t_p \bmod Q$. On the other hand, for any $\mathbf{z} \in \mathbb{Z}^w$, the error $e \in \mathbb{Z}_p$ satisfying $\langle \mathbf{f}, \mathbf{z} \rangle = t - e \bmod Q$ is unique, i.e. $e = t_p$. Then $\langle \mathbf{f}, \mathbf{z} \rangle = t - t_p \bmod Q$ holds if and only if $\mathbf{z} \in \Lambda_{t_q}^{\perp}(\mathbf{g}^t)$. Therefore, the distribution of $\mathbf{z}$ is exactly $D_{\mathbb{Z}^w, r}$ conditioned on $\langle \mathbf{f}, \mathbf{z} \rangle = t - t_p \bmod Q$ for some $e \in \mathbb{Z}_p$.

Remark 1. In step 2 of Algorithm 1, we can sample $\mathbf{z} \leftarrow D_{\Lambda_{t_q}^{\perp}(\mathbf{g}^t), r}$ by using the techniques in [GPV08, MP12, GM18, HJ19, ZY22]. In SRNSG the parameter q is a power of b, i.e., $q = b^w$, therefore we can sample $\mathbf{z}$ with great ease as in [MP12].

We now prove that for uniformly random target $t \in \mathbb{Z}_Q$, the preimage and error distributions of $\mathsf{ApproxGadget}(t, r, p, q)$ can be simulated.

Lemma 4. *Let $p, q > 0$ be integers, $Q = pq$, $r \geq \eta_\epsilon(\Lambda_q^{\perp}(\mathbf{g}^t))$ with some negligible $\epsilon > 0$. Then the following two distributions are statistically close.*

1. *First sample $t \leftarrow U(\mathbb{Z}_Q)$, then sample $\mathbf{z} \leftarrow \mathsf{ApproxGadget}(t, r, p, q)$, compute $e = t \bmod p$, output $(\mathbf{z}, t, e)$;*
2. *First sample $e \leftarrow U(\mathbb{Z}_p)$, then sample $\mathbf{z} \leftarrow D_{\mathbb{Z}^w, r}$, compute $t = e + \langle \mathbf{f}, \mathbf{z} \rangle \bmod Q$, output $(\mathbf{z}, t, e)$.*

Proof. The supports of two distributions are identical as follows:

$$\{(\mathbf{z}, t, e) \in \mathbb{Z}^w \times \mathbb{Z}_Q \times \mathbb{Z}_p \mid t = e + pz \bmod Q\}.$$

Distribution 1 outputs $(\mathbf{z}, t, e)$ with probability $P_1[(\mathbf{z}, t, e)] = \frac{1}{Q} \cdot P_1[\mathbf{z} \mid t] = \frac{1}{pq} \cdot \frac{\rho_r(\mathbf{z})}{\rho_r(\Lambda_{t_q}^{\perp}(\mathbf{g}^t))}$, and Distribution 2 with $P_2[(\mathbf{z}, t, e)] = \frac{1}{p} P_2[\mathbf{z} \mid e] = \frac{1}{p} \frac{\rho_r(\mathbf{z})}{\rho_r(\mathbb{Z}^w)}$. Since $r \geq \eta_\epsilon(\Lambda_q^{\perp}(\mathbf{g}^t))$ and $\rho_r(\mathbb{Z}^w) = \sum_{i \in \mathbb{Z}_q} \rho_r(\Lambda_i^{\perp}(\mathbf{g}^t))$, Lemma 2 shows $\rho_r(\Lambda_{t_q}^{\perp}(\mathbf{g}^t)) \in [\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}] \frac{\rho_r(\mathbb{Z}^w)}{q}$. Hence $P_1[(z, t, e)] \in [\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}] \cdot P_2[(z, t, e)]$ and we complete the proof.

3.2 Our improved preimage sampler

In this subsection, we describe a new preimage sampler based on the aforementioned gadget, together with the non-spherical Gaussian [JHT22]. Let $\Gamma = (n, m, p, q, Q, \chi)$ denote the global parameters where $Q = pq$ and χ is the distribution of secrets. Let $\mathbf{A} \in \mathbb{Z}_Q^{n \times m}$ be a matrix such that $m > n$. Our approximate trapdoor for $\mathbf{A}$ is defined as a matrix $\mathbf{T} \in \mathbb{Z}^{m \times n}$ such that $\mathbf{A} \cdot \mathbf{T} = \mathbf{F} = \mathbf{I}_n \otimes \mathbf{f} \bmod Q$, where $\mathbf{f} = p \cdot \mathbf{g} = p \cdot (1, b, \ldots, b^{w-1})$ is the gadget vector. The quality of the trapdoor is measured by its largest singular value $s_1(\mathbf{T})$. Similar to [MP12, CGM19], our trapdoor can be instantiated in statistical mode or computational mode, for higher efficiency, we only consider the computational mode in this work.

Let $\varSigma = \sigma_1^2 \cdot \mathbf{I}_{2n} \oplus \sigma_2^2 \cdot \mathbf{I}_{wn}$ and $\varSigma_{\mathbf{p}} = \varSigma - r^2 \cdot \mathbf{T} \cdot \mathbf{T}^t$. Algorithm 2 illustrates the preimage sampling algorithm by using the aforementioned approximate gadget trapdoor and the non-spherical technique in [JHT22]. At a high level, the sampling procedure follows the same manner with [MP12, CGM19] and uses the gadget sampler as "black-box". The output $\mathbf{x}$ satisfies

$$\mathbf{A}\mathbf{x} = \mathbf{F}\mathbf{z} + \mathbf{A}\mathbf{p} = \mathbf{v} - \mathbf{e} + \mathbf{A}\mathbf{p} = \mathbf{u} - \mathbf{e} \bmod Q.$$

Thus the approximation error $\mathbf{e}$ in Algorithm 2 is exactly the one in Algorithm 1, i.e., for uniformly random $\mathbf{u}$, the error $\mathbf{e}$ is uniformly random over $\mathbb{Z}_p^n$.

Algorithm 2: $\mathsf{ApproxPreSamp}(\mathbf{A}, \mathbf{T}, \mathbf{u}, r, \varSigma)$

Require: $(\mathbf{A}, \mathbf{T}) \in \mathbb{Z}_Q^{n \times m} \times \mathbb{Z}^{m \times wn}$ such that $\mathbf{A}\mathbf{T} = \mathbf{F} \bmod Q$, a vector $\mathbf{u} \in \mathbb{Z}_Q^n$, $r \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{g}^t))$ and $\varSigma$ such that $\varSigma_{\mathbf{p}} \succ \mathbf{0}$

Ensure: an approximate preimage $\mathbf{x}$ of $\mathbf{u}$ for $\mathbf{A}$.

1: $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\varSigma_{\mathbf{p}}}}$

2: $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \bmod Q$

3: $\mathbf{z} \leftarrow \mathsf{ApproxGadget}(\mathbf{v}, r, p, q)$

4: **return** $\mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z}$

Let $\mathbf{L} = [\mathbf{I}_m \mid \mathbf{T}]$. The next lemma characterizes the distribution of the linear transformation on the concatenation of $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\varSigma_{\mathbf{p}}}}$ and $\mathbf{z} \leftarrow D_{\mathbb{Z}^n, r}$, which represents the convolution step, i.e.,

$$\mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z} = \mathbf{L} \cdot (\mathbf{p}, \mathbf{z}).$$

Lemma 5 ([JHT22], adapted). *Let $\varSigma = \sigma_1^2 \cdot \mathbf{I}_{2n} \oplus \sigma_2^2 \cdot \mathbf{I}_{wn}$. For $\sigma_1^2 \geq (r^2 + \bar{r}^2) \cdot \left(s_1(\mathbf{T})\right)^2 + 2r^2 + 4\bar{r}^2$ and any σ_2^2 such that $\varSigma_{\mathbf{p}} \geq \bar{r}^2$, the distribution $\mathbf{L} \cdot D_{\mathbb{Z}^{m+wn}, \sqrt{\varSigma_{\mathbf{p}} \oplus r^2 \cdot \mathbf{I}_{wn}}}$ is statistically close to $D_{\mathbb{Z}^m, \sqrt{\varSigma}}$.*

Now we are ready to present the main theorem to state that the preimage and error distributions are simulatable without knowing the trapdoor, for uniformly random target $\mathbf{u}$, as in [CGM19]. The proof follows that in [CGM19, JHT22], but is slightly simpler, as we only use Theorem 1 once instead of twice.

Theorem 2. *Let $(\mathbf{A}, \mathbf{T})$ be a matrix-approximate trapdoor pair, $\mathbf{B} = \begin{bmatrix} \mathbf{T} \\ -\mathbf{I}_n \end{bmatrix}$ and $(r, \varSigma)$ such that $\sqrt{\varSigma_{\mathbf{p}} \oplus r^2 \mathbf{I}_n} \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$. Denote by $\mathbf{A}^{-1}(\cdot)$ the shorthand of*

$\mathsf{ApproxPreSamp}(\mathbf{A}, \mathbf{T}, \cdot, r, \Sigma)$. *Then the following two distributions are statistically indistinguishable:*

$$\left\{(\mathbf{A}, \mathbf{x}, \mathbf{u}, \mathbf{e}) : \ \mathbf{u} \leftarrow U(\mathbb{Z}_Q^n), \ \mathbf{x} \leftarrow \mathbf{A}^{-1}(\mathbf{u}), \ \mathbf{e} = \mathbf{u} - \mathbf{A}\mathbf{x} \bmod Q\right\}$$
$$\left\{(\mathbf{A}, \mathbf{x}, \mathbf{u}, \mathbf{e}) : \ \mathbf{x} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma}}, \ \mathbf{e} \leftarrow U(\mathbb{Z}_p^n), \ \mathbf{u} = \mathbf{A}\mathbf{x} + \mathbf{e} \bmod Q\right\}$$

Proof. Let

- $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_\mathbf{p}}}$ be a perturbation,
- $\mathbf{u} \in \mathbb{Z}_Q^n$ be the input target,
- $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p} \bmod Q$ be the target of the algorithm $\mathsf{ApproxGadget}(\mathbf{v}, r, p, q)$.

Real distribution: The real distribution of $(\mathbf{A}, \mathbf{x}, \mathbf{u}, \mathbf{e})$ is

$$\mathbf{A}, \mathbf{u} \leftarrow U(\mathbb{Z}_Q^n), \mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_\mathbf{p}}}, \mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p},$$
$$\mathbf{z} \leftarrow \mathsf{ApproxGadget}(\mathbf{v}, r, p, q), \mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z}, \mathbf{e} = \mathbf{u} - \mathbf{A}\mathbf{x}.$$

Hybrid 1: Instead of sampling $\mathbf{u} \leftarrow U(\mathbb{Z}_Q^n)$, we sample $\mathbf{v} \leftarrow U(\mathbb{Z}_Q^n)$ and $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_\mathbf{p}}}$, and compute $\mathbf{u} = \mathbf{v} + \mathbf{A}\mathbf{p}$. We keep $(\mathbf{z}, \mathbf{e}, \mathbf{x})$ unchanged. Clearly, the real distribution and Hybrid 1 are the same.

Hybrid 2: Instead of sampling $\mathbf{v}, \mathbf{z}$ and computing $\mathbf{e}$ as in Hybrid 1, we sample $\mathbf{z} \leftarrow D_{\mathbb{Z}^{wn}, r}$ and $\mathbf{e} \leftarrow U(\mathbb{Z}_p^n)$, and compute $\mathbf{v} = \mathbf{e} + \mathbf{F}\mathbf{z}$. All other terms $(\mathbf{p}, \mathbf{x}, \mathbf{u})$ remain unchanged. By Lemma 4, Hybrid 1 and Hybrid 2 are statistically close.

Hybrid 3: Instead of sampling $\mathbf{p}, \mathbf{z}$ and compute $\mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z}$ in Hybrid 2, we sample directly $\mathbf{x} \leftarrow D_{\mathbb{Z}^m, s}$ and compute $\mathbf{u} = \mathbf{A}\mathbf{x} + \mathbf{e}$. Note that in Hybrid 2,

$$\mathbf{u} = \mathbf{v} + \mathbf{A}\mathbf{p} = \mathbf{e} + p\mathbf{z} + \mathbf{A}\mathbf{p} = \mathbf{e} + \mathbf{A}(\mathbf{p} + \mathbf{T}\mathbf{z}) = \mathbf{A}\mathbf{x} + \mathbf{e} \bmod Q$$

and $\mathbf{x} = \mathbf{p} + \mathbf{T}\mathbf{z}$ follows the distribution $[\mathbf{I}_m \mid \mathbf{T}] \cdot D_{\mathbb{Z}^{m+n}, \sqrt{\Sigma_\mathbf{p} \oplus r^2 \mathbf{I}_n}}$. By Lemma 5, Hybrid 3 and Hybrid 2 are statistically close. This completes the proof.

4 Specification of the Optimized IBE

This section gives a complete specification of the SRNSG IBE algorithm. We first summarize the parameters and notations in Table 2. For ease of notation, we treat the each element $f \in \mathcal{K}$ and its matrix form $\mathcal{M}(f)$ as identical.

4.1 Setup

SRNSG uses the RLWE-style key pair. Its master secret key is $\mathbf{R} \leftarrow \chi_n^{2 \times w}$ and the master public key is essentially $(a, \mathbf{b} = \mathbf{f} - [1, a] \cdot \mathbf{R} \bmod Q)$, where $\mathbf{f}$ is the gadget. A formal description of the key generation is given in Algorithm 3.

The element $a \in \mathcal{R}_Q$ is generated by an ideal extendable-output function XOF with a 32-byte seed seed_a. For compactness, it is stored as seed_a.

Table 2. Description of parameters and notations.

	Description
l_m	message length, $l_m = 256$
l_s	seed length, $l_s = 256$
l_v	codeword length $l_v = 511$
l_t	ECC codeword distance
(p, q)	gadget parameters
Q	global modulus, $Q = pq$
b	a small integer as the log base
w	dimension of gadget vector $\mathbf{f}$, $w = \lceil \log_b q \rceil$
n	a power of 2 integer
$\mathcal{R}$	$\mathbb{Z}[X]/(X^n + 1)$
$\mathcal{R}_Q$	$\mathcal{R}/(Q \cdot \mathcal{R})$
χ	centered binomial distribution with parameter 1/2, i.e., $\{-1 : \frac{1}{8}, 0 : \frac{3}{4}, 1 : \frac{1}{8}\}$
χ_n	centered binomial distribution over $\mathcal{R}$ with parameter 1/2
Q_s	upper bound of extraction query number, $Q_s = 2^{30}$
ϵ	closeness parameter, $\epsilon = 1/\sqrt{Q_s \cdot 256}$
$\bar{r}$	base Gaussian parameter $\bar{r} = \eta_\epsilon \left(\mathbb{Z}^{(2+w)\cdot 2048}\right)$
r	gadget Gaussian parameter $r = b\bar{r}$ by lemma 1
σ_1, σ_2	standard deviation of the preimage
$\mathbf{\Sigma}$	covariance matrix of the preimage, $\mathbf{\Sigma} = \sigma_1^2 \cdot \mathbf{I} \oplus \sigma_2^2 \cdot \mathbf{I}$
$\mathbf{f}$	gadget vector $\mathbf{f} = p \cdot [1, b, \ldots, b^{w-1}] \in \mathcal{R}^w$
$\mathbf{R}$	secret matrix $\mathbf{R} \leftarrow \chi_n^{2\times w}$
$(a, \mathbf{b})$	public elements $a \leftarrow U(\mathcal{R})$, $\mathbf{b} = \mathbf{f} - [1,\ a] \cdot \mathbf{R} \bmod Q$
$\mathbf{\Sigma_p}$	perturbation covariance $\mathbf{\Sigma_p} = \mathbf{\Sigma} - r^2 \cdot \left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] \cdot [\mathbf{R}^t\ \mathbf{I}] = \left[\begin{smallmatrix}\sigma_1^2\cdot\mathbf{I} - r^2\mathbf{R}\mathbf{R}^t & -r^2\mathbf{R}\\ -r^2\mathbf{R}^t & (\sigma_2^2 - r^2)\cdot\mathbf{I}\end{smallmatrix}\right] \succ \bar{r}^2$
$\mathbf{\Sigma}_2$	the Schur complement of $(\sigma_2^2 - r^2)\cdot\mathbf{I}$ in $\mathbf{\Sigma_p}$, i.e., $\mathbf{\Sigma}_2 = \sigma_1^2\cdot\mathbf{I} - \frac{\sigma_2^2\cdot r^2}{\sigma_2^2 - r^2}\mathbf{R}\mathbf{R}^t = \left[\begin{smallmatrix}a & b\\ b^* & d\end{smallmatrix}\right]$
$\mathbf{C}$	$\mathbf{C} = \left[\begin{smallmatrix}a - bd^{-1}b^* & b\\ & d\end{smallmatrix}\right]$

In step 3 to step 5 of Algorithm 3, we need to sample the trapdoor $\mathbf{R}$ such that $\mathbf{\Sigma_p} - \bar{r}^2 \cdot \mathbf{I}$ is positive definite. This can be realized by: (1) checking the positive definiteness of the Schur complements of sub-matrices in $\mathbf{\Sigma_p}$ recursively; then (2) checking the definiteness of ring elements in $\mathcal{K}$. In more detail, since $\sigma_2 > r$ in our parameters, we need to check the positive definiteness of the Schur complement $\mathbf{\Sigma}_2$ of $(\sigma_2^2 - r^2) \cdot \mathbf{I}$, which in turn follows this procedure and boils down to check the positive definiteness of field elements in $\mathcal{K}_{\mathbb{R}}$. To this end, we simply compute their canonical embedding respectively, then check whether each element is positive or not.

In step 7, the element $a - bd^{-1}b^*$ in $\mathbf{C}$ is the Schur complement of d in $\mathbf{\Sigma}_2$. We include the triangular matrix $\mathbf{C}$ as a part of the secret key to simplify the key extraction procedure.

Algorithm 3: Setup

Require: None
Ensure: $(mpk,\ msk)$
1: $\mathsf{seed}_a \leftarrow \{0,1\}^{l_s}$, $a \leftarrow \mathsf{XOF}(\mathsf{seed}_a)$ ▷ $a \in \mathcal{R}_Q$
2: **repeat**
3: $\quad \mathbf{R} \leftarrow \chi_n^{2\times w}$ ▷ $\mathbf{R} \in \mathcal{R}^{2\times w}$
4: $\quad \mathbf{\Sigma_p} = \begin{bmatrix} \sigma_1^2 \cdot \mathbf{I} & \\ & \sigma_2^2 \cdot \mathbf{I} \end{bmatrix} - r^2 \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \cdot [\mathbf{R}^t\ \mathbf{I}]$
5: **until** $\mathbf{\Sigma_p} \succ \bar{r}^2$
6: Let $\mathbf{\Sigma}_2 = \sigma_1^2 \cdot \mathbf{I} - \frac{r^2}{r^2 - \bar{r}^2} \mathbf{R}\mathbf{R}^t = \begin{bmatrix} a & b \\ b^* & d \end{bmatrix}$
7: $\mathbf{C} = \begin{bmatrix} a - bd^{-1}b^* & b \\ & d \end{bmatrix}$ ▷ $\mathbf{C} \in \mathcal{K}_{\mathbb{R}}^{2\times 2}$
8: $\mathbf{f} = p \cdot [1,\ b, \ldots,\ b^{w-1}]$ ▷ $\mathbf{f} \in \mathcal{R}^{1\times w}$
9: $\mathbf{b} = \mathbf{f} - [1,\ a] \cdot \mathbf{R} \bmod Q$ ▷ $\mathbf{b} \in \mathcal{R}^{1\times w}$
10: **return** $(mpk = (\mathsf{seed}_a, \mathbf{b}),\ msk = (\mathbf{R}, \mathbf{C}))$

4.2 Extract users' secret keys

On input a user's $id \in \{0,1\}^*$, the extracting procedure shown in Algorithm 4 produces a short preimage $\mathbf{y} = (y_0, \ldots, y_{w+1}) \in \mathcal{R}^{2+w}$ such that $[1, a, \mathbf{b}] \cdot \mathbf{y} = \mathsf{H}(id) - e \bmod Q$ for some small $e \in \mathcal{R}$. This procedure consists of two phases: offline and online, following the idea of [Pei10, MP12]. In the offline phase, it samples an integer perturbation vector $\mathbf{p}$ from $D_{\mathcal{R}^{2+w}, \mathbf{\Sigma_p}}$. Then in the online phase, it produces an approximate preimage using the semi-random sampling technique [YJW23], as shown in the previous section. The output secret key for each user is essentially $(\mathbf{y}_1, \ldots, \mathbf{y}_{w+1})$. For compactness, we use some encoding technique, like [ETWY22], to compress $(\mathbf{y}_1, \ldots, \mathbf{y}_{w+1})$.

The perturbation sampling algorithm is implemented with Peikert's Gaussian convolution technique [Pei10] at the ring level, together with Genise and Micciancio's technique [GM18] that samples perturbation $\mathbf{p}$ by gradually updating the center and the covariance matrix using the Schur complement. In more

Algorithm 4: Extract

Require: msk, $id \in \{0,1\}^*$
Ensure: sk_{id}
Offline phase:
1: $\mathbf{p} \leftarrow \mathsf{SampleP}(msk)$ ▷ $\mathbf{p} \in \mathcal{R}^{2+w}$
2: $a \leftarrow \mathsf{XOF}(\mathsf{seed}_a)$
3: $u = H(id)$ ▷ $u \in \mathcal{R}_Q$
4: $v = u - [1, a, \mathbf{b}] \cdot \mathbf{p} \mod Q$ ▷ $v \in \mathcal{R}_Q$
Online phase:
5: $\mathbf{x} \leftarrow \mathsf{SampleGadgetF}(v)$ ▷ $\mathbf{f} \cdot \mathbf{x} \approx v \bmod Q$
6: $\mathbf{y} = \mathbf{p} + \left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] \cdot \mathbf{x}$ ▷ $\mathbf{y} = (y_0, y_1, \ldots, y_{w+1}),\ [1, a, \mathbf{b}] \cdot \mathbf{y} \approx u \bmod Q$
7: **return** $sk_{id} = (y_1, \cdots, y_{w+1})$ ▷ $[a, \mathbf{b}] \cdot sk_{id} \approx u - y_0 \bmod Q$

detail, $\mathsf{SampleP}$ proceeds as follows. First, it samples $\mathbf{p}' = (p_2, \ldots, p_{w+1}) \in \mathcal{R}^w$ with variance $\sigma_2^2 - r^2$. Then by [GM18], it samples $(p_0, p_1) \in \mathcal{R}^2$ with covariance $\mathbf{\Sigma}_2 = \sigma_1^2 \cdot \mathbf{I} - \frac{\sigma_2^2 \cdot r^2}{\sigma_2^2 - r^2} \mathbf{R}\mathbf{R}^t = \left[\begin{smallmatrix} a & b \\ b^* & d \end{smallmatrix}\right]$ and center $\mathbf{c} = -\frac{r^2}{\sigma_2^2 - r^2} \cdot \mathbf{R} \cdot \mathbf{p}' = (c_0, c_1)$. To achieve this, it continues the above recursive procedure, i.e., 1. sample p_1 with covariance d and center c_1; 2. sample p_0 with updated covariance $a - bd^{-1}b^*$ and center $c_0 + (p_1 - c_1)bd^{-1}$. Notice that the above procedure can be adapted recursively to sampling over $\mathcal{R}$ by exploiting the tower structures of $\mathcal{R}$ and $\mathcal{K}$, as shown in [GM18]. However, to avoid sampling over $\mathbb{Z}$ with large and varying variance, which has great impact on time efficiency and side-channel security, we use Peikert's Gaussian convolution technique [Pei10] at the ring level to keep efficiency. That is, given covariance $d \in \mathcal{K}_{\mathbb{R}}$ and center $c \in \mathcal{K}_{\mathbb{R}}$, it first samples a continuous Gaussian vector y of covariance $d - \bar{r}^2$, which can be done by applying the linear transformation defined by the Gram root of $d - \bar{r}^2$. Then it rounds the real coefficients of $y + c$ to some near integer by the integer Gaussian sampler $\mathsf{SampleZ}$ (Algorithm 7). The detailed algorithm is shown in Algorithm 5.

Algorithm 5: SampleP

Require: msk
Ensure: $\mathbf{p} \sim D_{\mathcal{R}^{2+w}, \sqrt{\mathbf{\Sigma_p}}}$
1: $\mathbf{p}' \leftarrow D_{\mathbb{Z}^{w \cdot n}, \sqrt{\sigma_2^2 - r^2}}$ ▷ $\mathbf{p}' \sim D_{\mathcal{R}^w, \sqrt{\sigma_2^2 - r^2}}$
2: $\mathbf{c} = (c_0, c_1) = \frac{r^2}{\sigma_2^2 - r^2} \cdot \mathbf{R} \cdot \mathbf{p}'$ ▷ $\mathbf{c} \in \mathcal{K}_{\mathbb{R}}^2$
3: $\mathbf{\Sigma}_2 = \sigma_1^2 \cdot \mathbf{I} - \frac{\sigma_2^2 \cdot r^2}{\sigma_2^2 - r^2} \mathbf{R}\mathbf{R}^t = \left[\begin{smallmatrix} a & b \\ b^* & d \end{smallmatrix}\right]$
4: $p_1 \leftarrow \mathsf{SampleFz}(d, c_1)$
5: $p_0 \leftarrow \mathsf{SampleFz}(a - bd^{-1}b^*, c_0 + (p_1 - c_1)bd^{-1})$
6: **return** $\mathbf{p} = (p_0, p_1, \mathbf{p}')$

Algorithm 6 is simply a ring variant of Peikert's sampler [Pei10]. In step 3 we abuse the notation and it means that each coefficient c'_i of c' is used as input of $\mathsf{SampleZ}$, i.e., n independent parallel invocations of Algorithm 7.

Algorithm 6: $\mathsf{SampleFz}$

Require: a covariance d and a center c $\qquad \triangleright\ d, c \in \mathcal{K}_{\mathbb{R}}$

Ensure: $z \leftarrow D_{\mathcal{R},d,c}$

1: $y \leftarrow \mathcal{N}_n$

2: $c' = c + \sqrt{d - \bar{r}^2} \cdot y$ $\qquad \triangleright\ c' = c'_0 + c'_1 \cdot X + \cdots + c'_{n-1} \cdot X^{n-1} \in \mathcal{K}_{\mathbb{R}}$

3: $z \leftarrow \mathsf{SampleZ}(c')$ $\qquad \triangleright\ z \in \mathcal{R}$

4: **return** z

Algorithm 7 shows the sampler for $D_{\mathbb{Z},\bar{r},c}$ with arbitrary center $c \in \mathbb{R}$, which is adapted from [PFH+22, HPRR20]. It samples some fixed Gaussian using table-based approach (Algorithm 8) followed by a rejection sampling to make the output correct.

Algorithm 7: $\mathsf{SampleZ}$

Require: a center c

Ensure: $z \leftarrow D_{\mathbb{Z},\bar{r},c}$

$d \leftarrow c - \lfloor c \rceil$

$z^+ \leftarrow \mathsf{BaseSample}()$

$b \leftarrow U(\{0, 1\})$

$z \leftarrow b + (2b - 1)z^+$

$x \leftarrow \frac{(z-d)^2 - (z^+)^2}{2\bar{r}^2}$

$r \leftarrow U(\{0, 1, \ldots, 2^{64} - 1\})$

if $r > \exp(x)$ **then**

  restart

end if

return $z + \lfloor c \rceil$

In step 4 of Algorithm 8, RCDT means the reverse cumulative distribution table with size 13, similar to that in Falcon [PFH+22], according to which one can sample a non-negative integer efficiently.

Algorithm 9 consists mainly of n parallel approximate gadget sampling and outputs a vector $\mathbf{x} \in \mathcal{R}^w$ such that $\mathbf{x}$ is an approximate image under the gadget $\mathbf{f} \in \mathcal{R}^w$. This is a concrete specification of our gadget sampler presented in Section III. Notice that in step 4, $\mathbf{z}_i$'s form a w-by-n matrix and each row of the matrix is converted naturally to a ring element by the coefficient embedding.

Given a target $u \in \mathbb{Z}_Q$, Algorithm 10 samples an approximate $\mathbf{z}$ such that $\langle p \cdot \mathbf{g}, \mathbf{z} \rangle = u - e \mod Q$ for some small $e \in \mathbb{Z}_p$, where $\mathbf{g} = (1, b, \ldots, b^{w-1}) \in \mathbb{Z}^w$.

Algorithm 8: BaseSample

Require: None
Ensure: $z^+ \leftarrow D^+_{\mathbb{Z},\bar{r}}$
1: $u \leftarrow U(\{0,1\}^{72})$
2: $z^+ \leftarrow 0$;
3: **for** $i = 0, \ldots, 12$ **do**
4: $\quad z^+ \leftarrow z^+ + [\![u < \mathrm{RCDT}[i]]\!]$
5: **end for**
6: **return** z^+

Algorithm 9: SampleGadgetF

Require: $v \in \mathcal{R}_Q$ $\qquad \triangleright\ v(X) = v_0 + v_1 X + \cdots + v_{n-1}X^{n-1}$
Ensure: $\mathbf{x} \in \mathcal{R}^w$
1: **for** $i = 0$ to $n-1$ **do**
2: $\quad \mathbf{z}_i \leftarrow \mathsf{ApproxGadget}(v_i)$ $\qquad \triangleright\ \mathbf{z}_i \in \mathbb{Z}^w$
3: **end for**
4: $[\mathbf{z}_0, \ldots,\ \mathbf{z}_{n-1}] \Rightarrow (x_0, \ldots, x_{w-1}) = \mathbf{x} \in \mathcal{R}^w$
5: **return** $\mathbf{x}$ $\qquad \triangleright\ \mathbf{f} \cdot \mathbf{x} \approx v \bmod Q$

In step 1, the approximate error e is generated deterministically [YJW23], and the remaining steps follow the highly optimized gadget sampler (for modulus q being power-of-b) in [MP12], which consists of sampling and shift operations over integers. Notice that step 3 can be accomplished by calling Algorithm 7: $z_i \leftarrow b \cdot \mathsf{SampleZ}(-v/b) + v$.

Algorithm 10: ApproxGadget

Require: $u \in \mathbb{Z}_Q$
Ensure: $\mathbf{z} \in \mathbb{Z}^w$ $\qquad \triangleright\ \langle p \cdot \mathbf{g}, \mathbf{z}\rangle \approx u \mod Q$
1: $e = u \bmod p,\ v = (u-e)/p$
2: **for** $i = 0$ to $w-1$ **do**
3: $\quad z_i \leftarrow D_{b\mathbb{Z}+v,r}$
4: $\quad v = \frac{v - z_i}{b}$
5: **end for**
6: **return** $\mathbf{z} = (z_1, \ldots, z_w)$

4.3 Encryption

On input mpk, id and a message $\mu \in \{0,1\}^{l_m}$, the encryption procedure use the Dual-Regev encryption scheme [GPV08]. The subroutine ECCEnc converts the message μ into a codeword $\hat{\mu}$.

Algorithm 11: Enc

Require: mpk, id, μ
Ensure: ct
$a \leftarrow \mathsf{XOF}(\mathsf{seed}_a)$ ▷ $a \in \mathcal{R}_Q$
$u = H(id)$ ▷ $u \in \mathcal{R}_Q$
$\hat{\mu} = \mathsf{ECCEnc}(\mu)$ ▷ $\hat{\mu} \in \{0,1\}^{l_v}$
$s \leftarrow \chi_n, \mathbf{e} \leftarrow \chi_n^{1+w}, e \leftarrow \mathbb{Z}^{l_v}$ ▷ $s \in \mathcal{R},\ e \in \mathbb{Z}^{l_v}$
$\mathbf{c} = s \cdot [a, \mathbf{b}] + \mathbf{e} \bmod Q$ ▷ $\mathbf{c} \in \mathcal{R}_Q^{1\times(1+w)}$
$c = (s \cdot u)_{l_v} + e + \frac{Q}{2} \cdot \hat{\mu} \bmod Q$ ▷ $c \in \mathbb{Z}_Q^{l_v}$
return $ct = (\mathbf{c},\ c)$

4.4 Decryption

On input sk_{id}, ct, the decryption procedure first recovers the corresponding $\hat{\mu}$, then uses the subroutine ECCDec to decode it.

Algorithm 12: Dec

Require: sk_{id}, ct
Ensure: μ
$\widetilde{\mu} = c - (\mathbf{c} \cdot sk_{id})_{l_v}$
for $i = 0$ to $l_v - 1$ **do**
  if $\frac{Q}{4} \leq \widetilde{\mu}_i < \frac{3Q}{4}$ **then**
    $\hat{\mu}_i = 1$
  else
    $\hat{\mu}_i = 0$
  end if
end for
$\mu = \mathsf{ECCDec}(\hat{\mu})$
return μ

4.5 Recommended parameters

We specify three sets of parameter for the toy, NIST-1, NIST-5 security levels respectively in Table 3.

5 Security and Performance

5.1 Security

On the theoretical side, SRNSG follows the GPV-IBE construction [GPV08], which is secure under chosen-plaintext and chosen-identity attack assuming that

Table 3. Recommended parameters.

Security level	toy	NIST-1	NIST-5
Polynomial degree n	512	1024	2048
Modulus Q	98304	393216	1048576
Gadget parameters (p, q)	$(1536, 2^6)$	$(1536, 2^8)$	$(2^{12}, 2^8)$
Log base b	8	16	16
Gadget dimension w	2	2	2
Standard deviation σ_1	291.9	820.4	1159.9
Standard deviation σ_2	65.8	249.2	261.6
ℓ_2-norm of sk_{id}	14422.8	42967.3	96752.1
ECC codeword distance l_t	33	33	33
Single bit error rate	$2^{-11.3}$	$2^{-17.71}$	$2^{-21.9}$
Decryption error rate	$2^{-80.3}$	$2^{-190.6}$	$2^{-261.9}$
mpk size (in bytes)	2208	4896	10272
sk_{id} size (in bytes)	1696	4096	8320
ct size (in bytes)	4360	8510	16638

the LWE problem is hard, in the ROM [GPV08] or in the Quantum ROM (QROM) [BDF+11, KYY18]. We omit the details.

Concretely, we consider the cost of known lattice attacks and the estimation of concrete security following the core-SVP methodology [ADPS16]. We summarize the security estimation in Table 4. The details of concrete security estimate is shown in Supplementary Material A.

Table 4. The concrete security are estimated as the core-SVP hardness of known attacks.

Security level	toy	NIST-1	NIST-5
BKZ blocksize for primal attack	206	458	**1008**
Classical core-SVP security	60	133	**294**
Quantum core-SVP security	54	121	**267**
BKZ blocksize for dual attack	204	**458**	**1012**
Classical core-SVP security	**59**	**133**	295
Quantum core-SVP security	**54**	**121**	268

5.2 Performance

We provide a proof-of-concept implementation for x86 64 platform, written in standard C for both parameter sets. In this section, we report its performance.

The implementation is complied by gcc 8.3.0 and runs on Deepin 20.9. Table 5 shows the performance of our implementations on a single core of Intel Core i7-10710U @ 1.1 GHz with 8GB RAM.

Table 5. Performance of SRNSG. Numbers are the median cycle measured over $1,000$ executions.

Security level	toy	NIST-1	NIST-5
Setup	3,095,820	7,136,414	4,812,673
Extract	2,152,345	4,364,517	9,999,207
Enc	496,206	1,029,074	2,329,433
Dec	353,271	689,747	1,624,520

5.3 Comparison

We do not compare the implementations of IBE based on ideal lattice in [BFRLS18] and [BEP+21], as the different security models would make the comparison irrelevant. In contrast, in Table 6, we compare with the implementation based on NTRU lattice in [MSO17], which is the instantiation of the GPV-IBE as well. Notice that for a fair comparison, we re-estimate the security for their parameters in the core-SVP model.

As shown in Table 6, generally, the NTRU-based instantiation is more compact than SRNSG. For the NIST-I security level, while SRNSG has higher security level and much smaller *msk* size, the sizes of *mpk*, sk_{id} and *ct* are about 2 times the sizes of that in [MSO17]. This is an inherent gap, as the dimension of the underlying SIS problems instance is 2 times the dimension of that in [MSO17]. The higher dimension increases the sizes, although SRNSG uses smaller modulus.

However, it is worthy to note that SRNSG removes the NTRU assumption and its concrete security relies essentially on the Ring LWE assumption. This is an attractive feature especially for more powerful applications with overstretched parameters. Besides, SRNSG has significant advantages from the implementation standpoint: (1) SRNSG has very compact *msk* and avoids the notoriously complex NTRU trapdoor generation; (2) the offline phase of the extraction procedure in SRNSG can be more conveniently implemented without floating-point numbers [DGPY20]; (3) the base samplings of the online phase of the extraction procedure in SRNSG are in the form $D_{b\mathbb{Z}+z,r}$ for $z \in \mathbb{Z}$, which is beneficial for further optimization and side-channel protections. Finally, SRNSG can be more conveniently adapted to the unstructured setting, thanks to the absence of costly matrix inversions in the key generation.

Table 6. Comparison of SRNSG with [MSO17]. Sizes are in bytes. Note that here we only consider trapdoor size in the msk and omit the auxiliary matrix $\mathbf{C}$ in SRNSG or the Falcon tree in [MSO17].

Security level	toy	NIST-1	NIST-5
SRNSG Security C / Q	59 / 54	133 / 121	294 / 267
[MSO17] Security C / Q	43 / 39	122 / 111	—
SRNSG mpk size	2208	4896	10272
[MSO17] mpk size	1472	2944	—
SRNSG msk size	512	1024	2048
[MSO17] msk size	2208	4160	—
SRNSG sk_{id} size	1696	4096	8320
[MSO17] sk_{id} size	872	1744	—
SRNSG ct size	4350	8510	16638
[MSO17] size	1728	3584	—

6 Conclusion

We present the first instantiation of GPV-IBE on ideal lattices towards practical use. The main technique is an improved preimage sampling algorithm, which integrates a modified gadget sampler that is more suitable for practicalisation of advanced cryptographic schemes, and the non-spherical Gaussian technique [JHT22]. Besides, we provide two parameter sets and a proof-of-concept implementation. Thanks to the gadget structure, the key extraction procedure is easy and fast, which makes SRNSG an attractive post-quantum IBE for constrained environments. Given the importance of the preimage sampling algorithm in lattice-based cryptography, we believe that our technique can also be applied in the practicalisation of other advanced cryptographic schemes.

6.1 Future works

To support more flexible parameter choices, we can use the cyclotomic ring of 3-smooth conductor $m = 2^\ell \cdot 3^k$, instead of power-of-2 conductors. Alternatively, we may adapt the ring structure to the module setting, at the cost of increasing the master public key size.

It is worthy to implement our algorithms fully over integers by the techniques of [DGPY20] in the key extraction procedure, and [CHK+21] in the encryption and decryption algorithms, which supports NTT multiplication for our NTT-unfriendly modulus Q. We leave the optimized implementation as future works.

From the perspective of cryptographic functionality, we focus more on the basic IBE itself in this work. Actually, by using the FSXY [FSXY12] and FO [FO99, HHK17, JZC$^+$18] transformations, we can get efficient identity-based key exchange protocol in the CK$^+$ model and the identity-based KEM against the chosen ciphertext attack in the (Quantum) ROM.

Bibliography

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT 2010*, pages 553–572, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *CRYPTO 2010*, pages 98–115, 2010.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange–A New Hope. In *USENIX Security 2016*, pages 327–343, 2016.

[AFL16] Daniel Apon, Xiong Fan, and Feng-Hao Liu. Compact identity based encryption from LWE. Cryptology ePrint Archive, Report 2016/125, 2016.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, 2011.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA 2016*, pages 10–24, 2016.

[BEP$^+$21] Pauline Bert, Gautier Eberhart, Lucas Prabel, Adeline Roux-Langlois, and Mohamed Sabt. Implementation of lattice trapdoors on modules and applications. In *PQC 2021*, pages 195–214, 2021.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, pages 213–229, 2001.

[BFRLS18] Pauline Bert, Pierre-Alain Fouque, Adeline Roux-Langlois, and Mohamed Sabt. Practical implementation of ring-sis/lwe based signature and ibe. In *Post-Quantum Cryptography*, pages 271–291, 2018.

[BGK08] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In *ACM CCS 2008*, pages 417–426, 2008.

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous ibe, leakage resilience and circular security from new assumptions. In *EUROCRYPT 2018*, pages 535–564, 2018.

[BVWW16] Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring lwe. In *ITCS 2016*, pages 147–156, 2016.

[BWY11] Mihir Bellare, Brent Waters, and Scott Yilek. Identity-based encryption secure against selective opening attack. In *Theory of Cryptography*, pages 235–252, 2011.

[CGM19] Yilei Chen, Nicholas Genise, and Pratyay Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. In *ASIACRYPT 2019*, pages 3–32, 2019.

[CHK+21] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. Ntt multiplication for ntt-unfriendly rings: New speed records for saber and ntru on cortex-m4 and avx2. *IACR Transactions on CHES*, 2021, Issue 2:159–188, 2021.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT 2010*, pages 523–552, 2010.

[CN11] Yuanmi Chen and Phong Q Nguyen. Bkz 2.0: Better lattice security estimates. In *ASIACRYPT 2011*, pages 1–20, 2011.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding*, pages 360–363, 2001.

[DG17] Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In *CRYPTO 2017*, pages 537–569, 2017.

[DGPY20] Léo Ducas, Steven Galbraith, Thomas Prest, and Yang Yu. Integral Matrix Gram Root and Lattice Gaussian Sampling without Floats. In *EUROCRYPT 2020*, pages 608–637, 2020.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT 2014*, pages 22–41, 2014.

[DN12] Léo Ducas and Phong Q Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In *ASIACRYPT 2012*, pages 433–450, 2012.

[DP16] Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *ISSAC 2016*, pages 191–198, 2016.

[EFG+22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. MITAKA: A Simpler, Parallelizable, Maskable Variant of. In *EUROCRYPT 2022*, pages 222–253, 2022.

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter hash-and-sign lattice-based signatures. In *CRYPTO 2022*, pages 245–275, 2022.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *PKC 1999*, pages 53–68, 1999.

[FSXY12] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. In *PKC 2012*, pages 467–484, 2012.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO 1997*, pages 112–131, 1997.

[GM18] Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In *EUROCRYPT 2018*, pages 174–203, 2018.

[GMPW20] Nicholas Genise, Daniele Micciancio, Chris Peikert, and Michael Walter. Improved discrete gaussian and subgaussian analysis for lattice cryptography. In *PKC 2020*, pages 623–651, 2020.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206, 2008.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *STOC 2013*, pages 545–554, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from lwe. In *CRYPTO 2015*, pages 503–523, 2015.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *TCC 2017*, pages 341–371, 2017.

[HHP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: digital signatures using the NTRU lattice. In *CT-RSA 2003*, pages 122–140, 2003.

[HJ19] Yupu Hu and Huiwen Jia. A new gaussian sampling for trapdoor lattices with arbitrary modulus. Des. Codes Cryptogr., 2019.

[HPRR20] James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Isochronous gaussian sampling: From inception to implementation. In *PQC 2020*, pages 53–71, 2020.

[JHT22] Huiwen Jia, Yupu Hu, and Chunming Tang. Lattice-based hash-and-sign signatures using approximate trapdoor, revisited. IET Inf. Secur., 2022.

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive nizk proofs for linear subspaces. In *ASIACRYPT 2013*, pages 1–20, 2013.

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018*, pages 96–125, 2018.

[KY16] Shuichi Katsumata and Shota Yamada. Partitioning via non-linear polynomial functions: More compact ibes from ideal lattices and bilinear maps. In *ASIACRYPT 2016*, pages 682–712, 2016.

[KYY18] Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for gpv-ibe in the quantum random oracle model. In *ASIACRYPT 2018*, pages 253–282, 2018.

[Laa16] Thijs Laarhoven. *Search problems in cryptography.* PhD thesis, PhD thesis, Eindhoven University of Technology, 2016., 2016.

[LDK+22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. Dilithium: Submission to the NIST's post-quantum cryptography standardization process, 2022.

[Lew12] Allison Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT 2012*, pages 318–335, 2012.

[LLZ+18] Xianhui Lu, Yamin Liu, Zhenfei Zhang, Dingding Jia, Haiyang Xue, Jingnan He, Bao Li, and Kunpeng Wang. Lac: Practical ring-lwe based public-key encryption with byte-level modulus. Cryptology ePrint Archive, Paper 2018/1009, 2018.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, pages 700–718, 2012.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *SIAM Journal on Computing*, pages 372–381, 2004.

[MSO17] Sarah McCarthy, Neil Smyth, and Elizabeth O'Sullivan. A practical implementation of identity-based encryption over ntru lattices. In *Cryptography and Coding*, pages 227–246, 2017.

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *EUROCRYPT 2016*, pages 820–849, 2016.

[NR06] Phong Q Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In *EUROCRYPT 2006*, pages 271–288, 2006.

[Pei10] Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO 2010*, pages 80–97, 2010.

[PFH+22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Submission to the NIST's post-quantum cryptography standardization process, 2022.

[Pre15] Thomas Prest. *Gaussian Sampling in Lattice-Based Cryptography.* PhD thesis, PhD thesis, École Normale Supérieure Paris, 2015.

[SAB+20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, ancrède Lepoint, Vadim Lyubashevsky, John M. Schanc, Gregor Seiler, Damien Stehlé, and Jintai Ding. Kyber: Submission to the NIST's post-quantum cryptography standardization process, 2020.

[SE94] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66:181–199, 1994.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, pages 47–53, 1984.

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO 2009*, pages 619–636, 2009.

[Yam16] Shota Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In *EUROCRYPT 2016*, pages 32–62, 2016.

[YD18] Yang Yu and Léo Ducas. Learning strikes again: the case of the DRS signature scheme. In *ASIACRYPT 2018*, pages 525–543, 2018.

[YJW23] Yang Yu, Huiwen Jia, and Xiaoyun Wang. Compact lattice gadget and its applications to hash-and-sign signatures. CRYPTO 2023, 2023.

[ZY22] Shiduo Zhang and Yang Yu. Towards a simpler lattice gadget toolkit. In *PKC 2022*, pages 498–520, 2022.

Supplementary Material

A Concrete Security Estimates

Notice that the concrete security of SRNSG is related to the hardness of (the Ring variants of) LWE and SIS problems. However, since the SIS problem is much harder than the LWE problem for the same parameters in SRNSG, we omit the hardness estimation for the SIS problem, and only consider two embedding attacks that are commonly referred to as primal attack and dual attack for the LWE problem.

A.1 Lattice reduction and core-SVP hardness

The BKZ lattice reduction algorithm [SE94] and its optimized variants [CN11, MW16] are the best known algorithms for solving lattice problems. The BKZ algorithm can find short lattice vectors and this strength increases with the blocksize β of BKZ. For a d-dimensional lattice Λ, BKZ with blocksize β would find some short $\mathbf{v} \in \Lambda$ with

$$\|\mathbf{v}\| \leq \delta_\beta^d \mathsf{vol}(\Lambda)^{1/d} \quad \text{and} \quad \delta_\beta \approx \left(\frac{(\pi\beta)^{\frac{1}{\beta}}\beta}{2\pi e} \right)^{\frac{1}{2(\beta-1)}}$$

when $d > \beta > 50$.

The core-SVP methodology, proposed in [ADPS16], gives a common method to assess the cost of lattice attacks. Following this methodology, one first estimates the blocksize β required for successful attacks and then quantify the attack cost with the core-SVP hardness model that is conservative. Specifically, the cost of BKZ with blocksize β is estimated as $2^{0.292\beta}$[BDGL16] in the classical setting and $2^{0.265\beta}$[Laa16] in the quantum setting.

A.2 Primal and dual attack

The hardness of the LWE problem depends on the choices of n, Q and the standard deviation τ of the distribution χ for the trapdoor $\mathbf{R}$ (In SRNSG, $\tau = \frac{1}{2}$ is fixed). For primal attack, the LWE samples are converted to a unique-SVP instance for lattice Λ, then the BKZ algorithm is employed to recover the unique shortest vector $(\mathbf{s}, \mathbf{e}, 1)$, which consists of the LWE secret and error vectors. As shown in [ADPS16], the attack is successful if and only if

$$\|(\mathbf{s}, \mathbf{e}, 1)\| \sqrt{\frac{\beta}{l+n+1}} \leq \delta_\beta^{2\beta-(l+n+1)-1} \cdot Q^{\frac{l}{l+n+1}}.$$

where l denotes the number of LWE samples.

For dual attack, short vectors in the dual lattice are used to solve the decisional LWE problem. According to [ADPS16], if the BKZ algorithm is capable of finding a short vector of length $\alpha = \delta^{l+n} Q^{n/(l+n+1)}$, then one can break the decisional LWE problem with advantage $\epsilon = 4\exp(-2(\pi\alpha\tau/Q)^2)$.