A New Trapdoor over Module-NTRU Lattice and its Application to ID-based Encryption

Jung Hee Cheon[1], Duhyeong Kim[2], Taechan Kim[3], and Yongha Son[4]

[1] Seoul National University, Seoul, Korea
jhcheon@snu.ac.kr
[2] Seoul National University, Seoul, Korea
doodoo1204@snu.ac.kr
[3] NTT Secure Platform Laboratories, Japan
taechan.kim.ym@hco.ntt.co.jp
[4] Seoul National University, Seoul, Korea
emsskk@snu.ac.kr

Abstract. A trapdoor over NTRU lattice proposed by Ducas, Lyubashevsky and Prest (ASIACRYPT 2014) has been widely used in various crytographic primitives such as identity-based encryption (IBE) and digital signature, due to its high efficiency compared to previous lattice trapdoors. However, the most of applications use this trapdoor with the power-of-two cyclotomic rings, and hence to obtain higher security level one should double the ring dimension which results in a huge loss of efficiency.
In this paper, we give a new way to overcome this problem by introducing a generalized notion of NTRU lattices which we call *Module-NTRU* (MNTRU) lattices, and show how to efficiently generate a trapdoor over MNTRU lattices. Moreover, beyond giving parameter flexibility, we further show that the Gram-Schmidt norm of the trapdoor can be reached to about $q^{1/d}$, where MNTRU covers $d \geq 2$ cases while including NTRU as $d = 2$ case. Since the efficiency of trapdoor-based IBE is closely related to the Gram-Schmidt norm of trapdoor, our trapdoor over MNTRU lattice brings more efficient IBE scheme than the previously best one of Ducas, Lyubashevsky and Prest, while providing the same security level.

Keywords: SIS trapdoor; Module-NTRU lattice; Identity-based encryption

1 Introduction

In cryptography, a trapdoor is a special secret information which enables to compute the inverse of a function which is hardly to be done itself. Due to the strong property, trapdoor has been widely used in various fields of cryptography, especially to construct digital signature and public-key encryption. There also have been proposed several trapdoor constructions in lattice-based cryptography, which is receiving world-wide attention due to its quantum-resistance and worst-case to average-case reduction property [Pei16].

In [GPV08], Gentry, Peikert and Vaikuntanathan made one of the major breakthroughs in lattice-based trapdoor constructions: they showed how to efficiently generate a trapdoor of Short Integer Solution (SIS) function $f_{\mathbf{A}}(x) = \mathbf{A}\ x \pmod q$ for an $n \quad m$ random integer matrix $\mathbf{A}\ 2\ Z_q^{n\ \ m}$ $(q \quad 2)$. Later, Micciancio and Peikert [MP12] proposed more versatile trapdoor notion than that of [GPV08], and a new trapdoor of Learning with Errors (LWE) function $g_{\mathbf{A}}(s;e) = \mathbf{A}\ \ s+e \pmod q$.

These lattice trapdoors enable to construct first lattice-based hash-and-sign digital signatures and identity-based encryptions (IBE). However, these schemes devised with the lattice trapdoors are quite inefficient since they require the condition $m = \ (n \log q)$. This condition is induced from the *statistical property* of trapdoor constructions: In [GPV08], the trapdoor generation algorithm outputs a pair $(\mathbf{A};\mathbf{T})\ 2\ Z_q^{n\ \ m} \quad Z^{m\ \ m}$ where $\mathbf{T}$ is a trapdoor for the SIS function $f_{\mathbf{A}}$. To ensure the secrecy of $\mathbf{T}$, the distribution of $\mathbf{A}$ is set to be statistically close to the uniform distribution over $Z_q^{n\ \ m}$ which induces the large parameter $m = \ (n \log q)$.

To overcome this bottleneck, Ducas, Lyubashevsky and Prest [DLP14] introduced a variant of the SIS trapdoor proposed in [GPV08], which exploits *computational property* of NTRU [HHGP+03] instances instead of statistical property. For a polynomial ring $R := Z[X]=(\ (X))$ with some monic polynomial $\ (X)\ 2\ Z[X]$; a new trapdoor generation algorithm outputs a pair $h_{\mathrm{NTRU}}\ 2\ R_q$ and $\mathbf{T}_{\mathrm{NTRU}}\ 2\ R^{2\ \ 2}$ for $R_q := R=qR$ satisfying $(1; h_{\mathrm{NTRU}})\ \ \mathbf{T}_{\mathrm{NTRU}} = 0 \pmod q$: The most distinctive point is the secrecy of the trapdoor $\mathbf{T}_{\mathrm{NTRU}}$ is now obtained from the computational hardness assumption of NTRU lattices. Since the new trapdoor is based on polynomial ring and only requires $m = 2n$ (contrary to $m = \ (n \log q)$ before), it derives much better efficiency compared to the previous lattice trapdoors, and hence several practical cryptosystems has been proposed upon it; a practical lattice-based IBE scheme [DLP14] and a digital signature scheme Falcon [PAFZ19].

However, despite of the efficiency of NTRU trapdoor compared to the previous trapdoors, there still exist some issues related to parameters. Due to security and efficiency issues, the most popular and natural choice of monic polynomial $\ (X)$ is $X^n + 1$ with power-of-two n: First on the security side, it is known that NTRU instantiated with such polynomial has a provable security under appropriate parameter regimes [SS11], which means $\ (X) = X^n + 1$ with power-of-two n has more sound security ground than other choices. Of course this security reduction does not rule out the usability of other polynomials, but the more important reason lies on efficiency side; several implementation techniques [MSO17,PP19] are based on that choice of $\ (X)$; and especially the concrete analysis of [DLP14] also focuses on it.

However, this use of power-of-two n leads to some inflexibility on the optimized parameter selection for the desired security level. For example, [DLP14] analyzed that ring dimension $n = 512$ provides about 80-bit security, and the next ring dimension $n = 1024$ provides about 192-bit security. In this case, to obtain a moderate security level like 128; one is forced to use $n = 1024$ which actually provides much stronger security level, which leads to efficiency degradation. Regarding this problem, Falcon chooses non-power-of-two $n = 768$ and instantiate NTRU trapdoor with a polynomial ring $Z[X]=(X^n \quad X^{n=2} + 1)$; with considerable

efforts to boost efficiency of NTRU trapdoor over this ring. However, currently it cannot reach to power-of-two case performances, and requires too complicated implementation details.

Meanwhile, ring-LWE-based cryptosystems had also suffered from a similar problem, but they overcame this problem by introducing a general notion of ring-LWE so called module-LWE [LS15]. Roughly speaking, a ring-LWE sample consists of two elements in a polynomial ring R_q: Also for this case, $(X) = X^n + 1$ is the most popular choice and hence the total dimension $2n$ is also quite restrictive. Module-LWE solves this restriction by extending the concept of ring-LWE to $d(\quad 2)$ elements in a polynomial ring R_q: Clearly this yields the total dimension dn; and this enables one to use non-power-of-two total dimension such as 768 ($n = 256$, $d = 3$) and 1536 ($n = 512$, $d = 2$ or $n = 256$, $d = 5$), such as [BDK+18,DKL+18].

1.1 This Work

In this paper, we propose a generalized notion of NTRU lattices called module-NTRU(MNTRU) lattices which enables to solve the dimension inflexibility of NTRU-based cryptosystems. We also show efficient generation a trapdoor over MNTRU lattices, and argue that our generalization yields better efficiency than NTRU trapdoor as well as parameter flexibility. Based on our MNTRU trapdoor, we construct a new IBE scheme as a generalization of the Gentry-Peikert-Vaikuntanathan (GPV) framework [GPV08] based on NTRU trapdoor. We also rigorously analyze the parameter choices with respect to the correctness and the security of the scheme. Our generalization derives much efficient parameter instantiation upon previous IBE scheme over MNTRU lattices. Lastly, we provide a proof-of-concept implementation result on our IBE scheme as summarized in Table 1, which includes the comparison with the implementation result in [DLP14][5].

Generalization of NTRU trapdoor. As an analogue of generalization from ring-LWE to module-LWE, we generalize a context of NTRU lattices in R^2 to MNTRU lattices in a higher-dimensional R^d. See Figure 1 for the overview.

We first review the generation of trapdoor over NTRU lattices. One first randomly samples two small polynomial f and g; which we interpret this by sampling a matrix $\mathbf{S}_{\mathrm{NTRU}} := \begin{matrix} g \\ f \end{matrix} \quad 2\ R^2\ ^1$: Assuming f is invertible in R_q, the NTRU instance is defined by $h := g/f\ 2\ R_q$; where $(1; h)\ 2\ R_q^2$ satisfies

$$(1; h)\ \mathbf{S}_{\mathrm{NTRU}} \quad 0 \mod q:$$

[5] We remark that this implementation is literally for a proof-of-concept, and speed results should not be taken seriously. There has been many optimization techniques after [DLP14], and our implementation does not consider them.

[6] In [DLP14], this parameter set was claimed to have 192-bit security based on their own security analysis. However we adapt the latest, rather conservative security analysis of literature, and it concludes 87-bit security for that parameter set.

	[DLP14]	**Ours**
$(d, n, \log_2 q)$	(2, 1024, 26)	(3, 512, 19)
Bit-security	87[6]	147
Ciphertext size (bytes)	3328	2432
Master pk size (bytes)	3328	2432
User sk size (bytes)	2048	1152
User KeyGen (ms)	22.02	12.6
Enc + Dec (ms)	4.9	1.6

Table 1: Comparison between [DLP14] and our scheme. Both experiments are done on Intel (R) Xeon (R) Silver 4144 processor (2.20GHz CPU). Full implementation can be found on https://anonymous.4open.science/r/4a72c47d-d4d8-4034-9b5c-61f236e9942d/.

NTRU trapdoor ($d = 2$ case)

1. Sample a matrix $\mathbf{S}_{\text{NTRU}} \in \mathcal{R}^{2\ 1}$ having small entries.
2. Take a vector $\boldsymbol{a} = (a_1, a_2) \in \mathcal{R}^2$ such that $\boldsymbol{a} \cdot \mathbf{S}_{\text{NTRU}} = 0$.
3. Define an NTRU instance by
$$h = a_1^{\ 1} \cdot a_2 \in \mathcal{R}_q$$
4. Solve the NTRU equation to have $\boldsymbol{F} \in \mathcal{R}^2$ and completes NTRU trapdoor $\mathbf{T}_{\text{NTRU}} = [\mathbf{S}_{\text{NTRU}} || \boldsymbol{F}]$.

Hardness Assumption:

Hard to find $\mathbf{T}_{\text{NTRU}}$ from h

MNTRU trapdoor

1. Sample matrix $\mathbf{S}_{\text{MNTRU}} \in \mathcal{R}^{d\ (d\ 1)}$ having small entries.
2. Take a vector $\boldsymbol{a} = (a_1, \cdots, a_d) \in \mathcal{R}^d$ such that $\boldsymbol{a} \cdot \mathbf{S}_{\text{MNTRU}} = \mathbf{0}$.
3. Define an MNTRU instance by
$$\boldsymbol{h} = a_1^{\ 1} \cdot (a_2, \cdots, a_d) \in \mathcal{R}_q^{d\ \ 1}$$
4. Solve the MNTRU equation to have $\boldsymbol{F} \in \mathcal{R}^d$ and completes NTRU trapdoor $\mathbf{T}_{\text{MNTRU}} = [\mathbf{S}_{\text{MNTRU}} || \boldsymbol{F}]$.

Hardness Assumption:

Hard to find $\mathbf{T}_{\text{MNTRU}}$ from $\boldsymbol{h}$

Fig. 1: Overview of NTRU generalization

Here the NTRU lattice is defined by

$$_{\text{NTRU}} := \quad (u; v)\ 2\ R^2 : u + vh = 0 \mod q \quad ;$$

which can be understood as a integral lattice in Z^{2n} that contains an unusual short vector $(g; \quad f)$: Now one who knows f and g can generate a trapdoor basis $\mathsf{T}_{\text{NTRU}}\ 2\ Z^{2n\ \ 2n}$ of $_{\text{NTRU}}$ by finding $F; G\ 2\ R$ satisfying

$$gF \quad fG = q$$

so-called NTRU equation, with

$$\mathsf{T}_{\mathrm{NTRU}} = \begin{matrix} A(g) & A(G) \\ A(f) & A(F) \end{matrix}$$

where A represents the anti-circulant matrix transform of polynomials (for a detailed definition, see Section 2.1).

We generalize this framework for d 2 cases. For that, we first randomly sample a matrix $\mathsf{S}_{\mathrm{MNTRU}}$ 2 $R^{d\ (d\ 1)}$ where each component has small coefficients. Then we construct a vector $h_{\mathrm{MNTRU}} = (h_1; \quad ; h_d)$ 2 $R_q^{d\ 1}$ satisfying

$$(1; h_{\mathrm{MNTRU}})\ \mathsf{S}_{\mathrm{MNTRU}} \quad 0 \mod q;$$

from which we define a dn-dimensional MNTRU lattice

$$_{\mathrm{MNTRU};d} := \left((u_0; \quad ; u_{d\ 1}) \ 2\ R^d : u_0 + \sum_{i=1}^{\not{x}} u_i h_i = 0 \mod q \right) :$$

Such $h = (h_1; \quad ; h_{d\ 1})$ is determined by $\det_i$=$\det_1$ mod q where $\det_i$ is a determinant of $(d\ 1)$ $(d\ 1)$ submatrix of $\mathsf{S}_{\mathrm{MNTRU}}$: Then, to extend the matrix $\mathsf{S}_{\mathrm{MNTRU}}$ to a trapdoor $\mathsf{T}_{\mathrm{MNTRU}}$ 2 $R^{d\ d}$; we solve a generalized equation of fG $gF = q$, so-called the MNTRU equation

$$\sum_{i=1}^{\not{x}} \det_i \quad F_i = q:$$

Now, the solution of the MNTRU equation $F = (F_1; :::; F_d)^t$ 2 R^d completes the trapdoor

$$\mathsf{T}_{\mathrm{MNTRU}}\ 2\ Z^{dn\ \ dn} = A(\mathsf{S})jjA(F) :$$

For the IBE scheme construction in GPV framework, the Gram-Schmidt norm of $k\mathsf{T}_{\mathrm{MNTRU}}k$ plays a crucial role. In this point, we argue that with experimental verification, it can be reached to some small multiple of $q^{1=d}$; where the multiple constant c_d only depends on d: The effect of this Gram-Schmidt norm change is addressed in the next subsection.

Remark 1. In [CG17], the MNTRU equation appeared in the context of constructing NTRU-based *hierarchical IBE* (but not in a name MNTRU equation). However, we cannot find any complete paper except a slide lacking in details, and hence we cannot further compare this with our method.

IBE from MNTRU trapdoor in GPV framework. As we construct a generalized trapdoor $\mathsf{T}_{\mathrm{MNTRU}}$ for MNTRU lattice $_{\mathrm{MNTRU};d}$ for d 2; we can also apply the IBE construction of GPV framework, which includes [DLP14] as $d = 2$ case. The master key pair is $\mathsf{MPK} = h$ and $\mathsf{MSK} = \mathsf{T}_{\mathrm{MNTRU}}$; and the user key extract procedure outputs the last d 1 entries of a vector $s = (s_0; s_1; \quad ; s_{d\ 1})$ 2 R^d sampled from discrete Gaussian over a lattice $(\mathsf{T}_{\mathrm{MNTRU}})$: The encryption and

decryption can be understood as a module-LWE based encryption with secret key $s^0 = (1; s_1; \ ; s_{d-1})$; precisely, the ciphertext of a binary-coefficient $m \in R$ is given by

$$c = r \cdot (t; h_1; \ ; h_{d-1}) + e + \left\lfloor \frac{q}{2} \right\rfloor m; 0; \ ; 0$$

where $t \in R_q$ is a value that publicly computed from id; and r and $e = (e_0; \ ; e_{d-1})$ are randomly chosen element in R having ternary$(-1; 0)$ coefficients. The corresponding decryption is done by computing

$$\langle c; s^0 \rangle = \left\lfloor \frac{q}{2} \right\rfloor m + e_0 + rs_0 + \sum_{i=1}^{d-1} e_i s_i :$$

For correct decryption, modulus q should be set sufficiently large so that the total error term $e_0 + rs_0 + \sum_{i=1}^{d-1} e_i s_i$ is less than $q/4$. This is related to the size of s_i which is sampled from a discrete Gaussian over a lattice $\Lambda^\perp(\mathbf{T}_{\mathrm{MNTRU}})$ using $\mathbf{T}_{\mathrm{MNTRU}}$. Here, the minimal standard deviation σ of all known discrete Gaussian samplers [DP16] is proportional to the Gram-Schmidt norm $\|\mathbf{T}_{\mathrm{MNTRU}}\|$. Since we show that $\|\mathbf{T}_{\mathrm{MNTRU}}\|$ can be reached proportional to $q^{1-1/d}$; the total error size is reduced than NTRU ($d = 2$) case, and hence enables us to choose the smaller modulus q. This leads to overall decreases on key sizes and ciphertext sizes. We also give a proof-of-concept implementation based on the implementation of [DLP14], whose results can be found by Table 1.

We finally remark that, although our implementation shows better performance (speeds) than the implementation of [DLP14], the main point of our proposal should be understood as a new way to have better parameters, and implementation issues still remain as a future work. Indeed, there has been many optimization techniques that can be applied to NTRU setting like [MSO17,PAFZ19,PP19], and hence the current implementation of [DLP14] is much faster than its implementation of [DLP14]. It would be a clear future work to check whether such techniques are applicable to MNTRU case.

A Digital Signature Scheme. The key generation and extract procedure can be exploited as a hash-and-sign digital signature, which is a Falcon's design rationale. In this case, there is no additional error term that was required for encryption, and we do not have the modulus q size condition for correctness anymore. This makes one almost freely chooses the modulus q; and hence we cannot expect global improvement for this case. Indeed, as the public key consists of $d-1$ polynomials in R_q; and MNTRU trapdoor increases the public key size. Still, one may expect the signature size drop because it still proportional to $\|\mathbf{T}_{\mathrm{MNTRU}}\|$; which would be a glad change in the actual usage. However, as concrete parameter uses quite small q (such as 12289 or 18433 in Falcon), our generalization fails to bring the actual size improvement. For the detailed discussion, we refer Appendix B.

1.2 Roadmap

In Section 2, we define some notations and give preliminaries for understanding trapdoor over NTRU lattices. Then in Section 3 we introduce a general notion

of NTRU, what we called MNTRU, and discuss about the trapdoor. Based on trapdoor over MNTRU lattices, we construct an IBE scheme in Section 4.

2 Preliminaries

2.1 Notations

We will work in the ring $R := \mathbb{Z}[X]/(X^n + 1)$ and denote $R_q := R/qR = \mathbb{Z}_q[X]/(X^n + 1)$. Let $f := \sum_{i=0}^{n-1} f_i X^i$ and $g := \sum_{i=0}^{n-1} g_i X^i$ be polynomials in R_q where $f_i, g_i \in \mathbb{Z}_q$ for $0 \le i \le n-1$. Any polynomial operations in R_q are carried out modulo $X^n + 1$; for instance $f \cdot g$ (or fg) denotes the multiplication in R_q. We also denote the coefficient vector of $f \in R$ (resp. R_q or $\mathbb{R}[X]/(X^n+1)$) by $(f) \in \mathbb{Z}^n$ (resp. $\mathbb{Z}_q^n$ or $\mathbb{R}^n$). According to this, for a vector $\mathbf{f} = (f_1, \ldots, f_d) \in \mathbb{R}[X]/(X^n+1)^d$, we denote $\|\mathbf{f}\|$ be the Euclidean norm of $((f_1), \ldots, (f_d)) \in \mathbb{R}^{dn}$. For a finite set X, we denote by $x \xleftarrow{\$} X$ uniform randomly sampling x from X. We use ln to denote the natural logarithm and $\log_2$ to denote the binary logarithm.

For $f = \sum_{i=0}^{n-1} f_i X^i \in R$, we denote the n-dimensional anti-circulant matrix of f by

$$A_n(f) := \begin{pmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ f_{N-1} & f_0 & \cdots & f_{n-2} \\ \vdots & \ddots & \ddots & \vdots \\ f_1 & f_2 & \cdots & f_0 \end{pmatrix} = \begin{pmatrix} (f) \\ (xf) \\ \vdots \\ (x^{N-1} f) \end{pmatrix}.$$

We omit the subscript n when it is clear from context. The anti-circulant matrix preserves the addition and multiplication, that is, $A_n(f) + A_n(g) = A_n(f+g)$ and $A_n(f) \cdot A_n(g) = A_n(fg)$. The same notation applies for a vector $\mathbf{v}$ or a matrix $\mathbf{M}$ over R by component-wise manner.

For a $d \times d'$ matrix $\mathbf{M}$ over R, we denote an R-module generated by the columns of $\mathbf{M}$ by $\Lambda_R(\mathbf{M}) = \{\mathbf{M} \cdot \mathbf{x} : \mathbf{x} \in R^{d'}\}$. Note that this object can be viewed as an integral lattice $\Lambda(A_n(\mathbf{M}))$ in $\mathbb{Z}^{dn}$ generated by the columns of $A_n(\mathbf{M})$.

2.2 Gaussian Measures

For a full-rank n-dimensional lattice $\Lambda \subseteq \mathbb{R}^n$, *the discrete Gaussian distribution* with width $\sigma > 0$ and center $\mathbf{c} \in \mathbb{R}^n$ denoted by $D_{\Lambda,\sigma,\mathbf{c}}$ is a distribution over which samples $\mathbf{x} \in \Lambda$ with the probability

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) := \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{z} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{z})}$$

where $\rho_{\sigma,\mathbf{c}}(\mathbf{z}) := \exp\left(\frac{\|\mathbf{z} - \mathbf{c}\|^2}{2\sigma^2}\right)$.

There is an well-known parameter $\eta_\varepsilon(\Lambda)$ called *smoothing parameter* defined by [MR07], which is defined by the smallest $s > 0$ such that

$$\rho_{1/s,\mathbf{0}}(\Lambda^* \setminus \{\mathbf{0}\}) \le \varepsilon;$$

where is a dual lattice of : We also denote the scaled-version $\rho'_{\sqrt{}}(\ ) := \rho_{\frac{1}{2}\sqrt{}}(\ )$: In particular, it is known that from [GPV08]

$$\eta'_{\sqrt{}}(Z) \quad \frac{1}{} \quad \sqrt{\frac{1}{2}\ln\quad 2+\frac{2}{}} :$$

A Gaussian Sampler. An algorithm that approximately samples the discrete Gaussian is proposed by [GPV08], and we will use for our MNTRU lattices and IBE scheme. Here we omit the detail of the algorithm and simply de ne the syntax: for a basis B of a lattice L; we denote the [GPV08] algorithm that approximately samples $D_{\ ;\ ;\ c}$ by

$$\text{GaussianSample}_{\text{B}}(B\ ;\ ;\ c):$$

2.3 Kullback-Leibler Divergence

Instead of the traditional statistical distance concept to measure the distance of two distributions, we especially will make use of Kullback-Leibler divergence (or KL divergence) following the methodology of [DLP14].

Remark 2. In fact, the recent literature is using more general concept of distance called Renyi divergence, for example in [PAFZ19]. However, the previous work [DLP14] was analyzed with KL divergence, and hence in this paper we stick to the KL divergence for a clear comparison.

De nition 1 (Kullback-Leibler Divergence). Let P and Q be two distributions over a common countable set; and let S be the support of P: The Kullback-Leibler Divergence, noted D_{KL} of Q from P is de ned as:

$$D_{KL}(P\|Q) = \sum_{i2S} \ln\ \frac{P(i)}{Q(i)}\ \ P(i)$$

with the convention that $\ln(x=0) = +1$ for any $x > 0$:

It is known that, if two distribution P and Q has small KL divergence, hardness of any search problem that requires oracle queries for P is preserved even if the oracle queries is replaced with Q:

Lemma 1 (Lemma 1 of [PDG14]). Let A^P be an algorithm making at most q queries to an oracle sampling from a distribution P and returning a bit. Let $A\ \ 0$; and Q be a distribution such that $D_{KL}(P\|Q)\ \ "$: Let x (resp. y) denote the probability that A^P (resp. A^Q) outputs 1. Then,

$$|x\quad y| \quad \sqrt{\frac{q"}{2}}:$$

Finally, we have the following fact for KL divergence of the ideal discrete Gaussian and the Gaussian sampler that we will use.

Theorem 1 (Theorem 2 of [DLP14]). For any $\varepsilon \in (0, 1/4n)$, if $\eta_\varepsilon^0(\mathbb{Z}) \le B$, then

$$D_{KL}\left(D_{(B),\mathbf{c}} \,\|\, \text{GaussianSample}_{\mathbb{Z}}(B, \Sigma, \mathbf{c})\right) \le \frac{1+\varepsilon}{1-\varepsilon} - 1 \le 8n^2\varepsilon^2.$$

2.4 The NTRU Lattice

We recall the definition of the NTRU lattices.

Definition 2 (NTRU lattices). Let n be a power-of-two integer, and q be a positive integer. For $f, g \in R$, let $h = g/f \bmod q$. The NTRU lattice Λ_{NTRU} associated to h and q is

$$\Lambda_{NTRU} = \{(u, v) \in R^2 : u + vh = 0 \bmod q\}.$$

By the definition, Λ_{NTRU} can also be seen as a full-rank lattice in $\mathbb{Z}^{2n}$ generated by the columns of $A_{NTRU} = \begin{bmatrix} A_n(h) & qI_n \\ I_n & O_n \end{bmatrix}$.

Several cryptosystems that deal with the NTRU lattices base their security on the hardness assumption of the NTRU problem which states that if $f, g \in R_q$ are random small polynomials, their quotient g/f is indistinguishable from random in R_q.

An interesting aspect of the NTRU lattice is that it can be easily instantiated with a trapdoor basis. More precisely, as explained in [HHGP+03], one can find another basis by computing $F, G \in R$ such that $gF - fG = q$, and then a short trapdoor basis of Λ_{NTRU} is provided by the integral matrix

$$T_{NTRU} := \begin{bmatrix} A_n(g) & A_n(G) \\ A_n(f) & A_n(F) \end{bmatrix}.$$

3 Module-NTRU Lattices

In this section, we introduce the generalized notion of NTRU lattices described in Section 2.4. To give intuition, we understand the NTRU trapdoor generation by following. First, it samples short polynomials $f, g \in R$; and we view this by sampling a small matrix $S = \begin{bmatrix} g \\ f \end{bmatrix}$. Then, an NTRU instance $h = g/f \in R_q$ can be understood by an element obtained from a vector orthogonal to S. In this case, such orthogonal vector is clearly $(f, g) \in R^2$; and h comes from the quotient vector $(1, g/f) \in R_q^2$. Finally, we extend S to the trapdoor T_{NTRU} by solving the NTRU equation that satisfies $gF - fG = q$; and define $T_{NTRU} = \begin{bmatrix} A(g) & A(G) \\ A(f) & A(F) \end{bmatrix}$.

In Section 3.1, we elaborate the generalization of the above understanding of NTRU instance and trapdoor generation, which we call module-NTRU (MNTRU) instance and trapdoor. We will apply this new trapdoor for IBE scheme in later sections, and the Gram-Schmidt norm of the trapdoor matrix is closely related to its efficiency. Regarding this, we analyze and discuss about the Gram-Schmidt norm of the trapdoor matrix in Section 3.2.

3.1 Construction of MNTRU lattice and trapdoor

Our new construction essentially follows the above described framework for NTRU; we first set a small matrix $S \in R_q^{d \times (d-1)}$ which corresponds to $(g, f)^t$; and consider a vector orthogonal to S; say $\det = (\det_1, \ldots, \det_d)$ [7]. Then we define a MNTRU instance by a vector $h \in R_q^{d-1}$ such that $(1, h) = \det_1^{-1} \det$. Finally, we consider a generalized version of NTRU equation defined by

$$\sum_{i=1}^{d} \det_i \cdot F_i = q,$$

and by concatenating $F = (F_1, \ldots, F_d)$ to S, we complete the trapdoor T_{MNTRU} generation.

We elaborate from the generation of S: Firstly, we sample vector of polynomials $f_i = (f_{1,i}, \ldots, f_{d,i}) \in R^d$ for $1 \le i \le d-1$ where each $f_{j,i}$ is a small polynomial (having small coefficients), and define a matrix $S = [f_1, \ldots, f_{d-1}] \in R^{d \times (d-1)}$; and assume that S is full-rank in R_q which happens with high probability.

To find a vector orthogonal to S over R, we define S_i be the $(d-1) \times (d-1)$ matrix that results from deleting i-th row of S, and define $\det_i = (-1)^{i-1} \det(S_i)$. Then the following lemma holds.

Lemma 2. *The vector $\det = (\det_i)_{1 \le i \le d}$ satisfies $\det^t \cdot S = 0$ over R.*

Proof. We show $\det^t$ is orthogonal to each column f_i of S by considering a $d \times d$ matrix $M_i = [f_i \,||\, S]$. Since M_i has the same two columns, it has determinant 0. Now the cofactor expansion by the first column implies $\det(M_i) = \det^t \cdot f_i$, which ends proof. □

Assuming that $\det_1$ is invertible in R_q (hence S is full-rank in R_q), we define the MNTRU instance $h_{\mathrm{MNTRU}} \in R_q^{d-1}$ as

$$h_{\mathrm{MNTRU}} = (h_1, \ldots, h_{d-1}).$$

From Lemma 2, it holds that $(1, h_{\mathrm{MNTRU}}) \cdot S = 0 \bmod q$. We then define the dn-dimensional MNTRU lattice Λ_{MNTRU} associated to h and q by

$$\Lambda_{\mathrm{MNTRU}} = \{(u_0, \ldots, u_{d-1}) \in R^d : u_0 + u_1 h_1 + \cdots + u_{d-1} h_{d-1} = 0 \bmod q\},$$

whose basis is given by

$$A_{\mathrm{MNTRU}} := \begin{pmatrix} 0 & A(h_1) & A(h_2) & \cdots & A(h_{d-1}) & qI_n \\ I_n & O_n & O_n & & O_n & O_n \\ O_n & I_n & O_n & & O_n & O_n \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ O_n & O_n & & & I_n & O_n \end{pmatrix}.$$

[7] As its name indicates, this vector is indeed computed from the determinant of submatrices of S.

We proceed to the generation of the MNTRU trapdoor $T_{MNTRU} \in \mathbb{Z}^{dn \times dn}$ of $_{MNTRU}$: For that, we consider the generalized NTRU equation (MNTRU equation) which was previously defined in [PP19], where we utilize a restricted version: Given $S \in R^{d \times (d-1)}$, find polynomials $F_1, \ldots, F_d \in R$ such that

$$\sum_{i=1}^{d} \det_i(F_i) = q \tag{1}$$

where $\det_i$ for $1 \le i \le d$ are defined above. This can be done by generalizing the previous method in [HHGP+03], or applying more developed method of [PP19]. As our proof-of-concept implementation exploits the former method, we give the detailed procedure in Appendix A for completeness. For a solution vector $F = (F_1, \ldots, F_d) \in R^d$ of the MNTRU equation, we set the trapdoor $T_{MNTRU} \in \mathbb{Z}^{dn \times dn}$ as the concatenation of $A_n(S)$ and $A_n(F)$, i.e.,

$$T_{MNTRU} := (A_n(S) \| A_n(F)).$$

We know that $(1, h_{MNTRU}) \cdot S = 0 \bmod q$ from Lemma 2, and moreover (1) implies that $h(1, h_{MNTRU}) \cdot F^t = 0 \bmod q$; and hence a lattice $\Lambda(T_{MNTRU})$ is contained in $_{MNTRU}$: Finally, Lemma 3 below says $\Lambda(T_{MNTRU})$ is full-rank, which completes the construction of trapdoor T_{MNTRU} for the MNTRU lattice $_{MNTRU}$:

Lemma 3. $\det(\Lambda(T_{MNTRU})) = q^{dn}$.

Proof. We only need to show that $\det_R(S \| F) = q^d$: Let $e_i \in R^d$ denote the unit vector whose i-th component is 1 for $1 \le i \le d$. Since $\sum_{i=1}^{d} \det_i(F_i) = q$, the determinant of T_{MNTRU} is $(-1)^{d-1} q$. Let $M_{i,j}$ be the (i,j)-minor of T_{MNTRU}, the determinant of $(d-1) \times (d-1)$ matrix results from deleting i-th row and j-th column of T_{MNTRU}, and define $M_i := (M_{i,1}, M_{i,2}, \ldots, M_{i,d})^t \in R^d$. Then, by the cofactor expansion, it holds that

$$T_{MNTRU} \cdot M_i = (-1)^{i-1} \det(T_{MNTRU}) \cdot e_i = q e_i,$$

which proves our claim. □

Note that Lemma 3 only implies that $\Lambda(T_{MNTRU})$ is a full-rank sublattice of $_{MNTRU}$; but does not guarantee that $\Lambda(T_{MNTRU}) = _{MNTRU}$; and hence T_{MNTRU} is not proven to be a trapdoor basis for $_{MNTRU}$; recall that for NTRU case, T_{NTRU} is a basis of $_{NTRU}$: We first note that it is well known (e.g., Lemma 7.1 of [MG02]) that T_{MNTRU} can be efficiently converted into a basis B of $_{MNTRU}$ such that $\|B\| \le \|T\|$: As a more important remark, the full-rank set T_{MNTRU} indeed suffices for the trapdoor usage, and hence we never perform such basis-converting process in our IBE scheme.

Hardness Assumption. The original NTRU trapdoor obtains its hardness from NTRU assumption that as, for two small random polynomials f and g in R; their

quotient $h = fg^{-1} \in R_q$ is indistinguishable from uniform element in R_q: For our case, we can establish a similar MNTRU assumption, saying

$$h_{MNTRU} = det_1^{-1} \cdot (det_2, \ldots, det_d) \in R_q^{d-1}$$

is indistinguishable from a uniform vector in R_q^{d-1}:

In fact, what we exactly need is somewhat weaker notion; to apply the GPV framework, we require SIS is hard over a random choice of A: Thus, our following IBE scheme is secure under somewhat mild assumption that SIS is hard over A_{MNTRU} on average, where the randomness is from the random choice of f.

3.2 Minimize the Gram-Schmidt norm of T_{MNTRU}

For an IBE scheme in GPV framework, the users' secret key issue involves a discrete Gaussian sampling over (T_{MNTRU}): As known discrete Gaussian samplers sample Gaussian having size proportional to $\|\tilde{T}_{MNTRU}\|$; it is quite important to set T_{MNTRU} to have small Gram-Schmidt norm $\|\tilde{T}_{MNTRU}\|$: In this regard, we now explain how we choose $S \in R$ to minimize $\|\tilde{T}_{MNTRU}\|$:

We start from the following lemma adapted from Lemma 2 of [DLP14] that says for MNTRU trapdoor, we only need to seed Gram-Schmidt norms to determine $\|\tilde{T}_{MNTRU}\|$:

Lemma 4. Let $T_{MNTRU} = [t_1 \cdots t_{dn}]$ be the MNTRU trapdoor. Then

$$\|\tilde{T}_{MNTRU}\| = \max\{\|t_1\|, \|t_{n+1}\|, \ldots, \|t_{(d-1)n+1}\|\}$$

Intuitively, we expect that the minimal occurs when

$$\|t_1\| = \|t_{n+1}\| = \cdots = \|t_{(d-2)n+1}\| = \|t_{(d-1)n+1}\|:$$

Since the first $d-1$ norms depend on our choice of $f_i \in R^d$; we first choose $f_{i+1} \in R^d$ for $1 \le i \le d-2$ for the first $d-2$ equality by

$$\|t_{in+1}\| = \sqrt{\frac{d}{d-i}}\,\|t_1\|: \tag{2}$$

As underlying idea for this choice, we see that t_{in+1} is a projection of t_{in+1} (of dimension dn) over a subspace of dimension $(d-i)n$; and hence random choice of f_i implies

$$\|t_{(i-1)n+1}\| = \sqrt{\frac{d-i+1}{d}}\,\|t_{(i-1)n+1}\|:$$

We experimentally check this choice of f_i indeed implies

$$\|t_1\| = \|t_{n+1}\| = \cdots = \|t_{(d-2)n+1}\|;$$

and Figure 2 in Appendix C shows the result with $d = 4$ case.

Finally the last one $\|t_{(d-1)n+1}\|$ depends on our choice of $S = [f_1, \ldots, f_{d-1}]$; and we investigate the optimal choice of $\|t_1\|$ while varying $\|t_1\|$: We presume that

such optimal choice is represented by $c_d \cdot q^{1-d}$ for some constant c_d that depends only on d; which implies the Gram-Schmidt norm of T_{MNTRU} can be reached to

$$\|\tilde{T}_{MNTRU}\| \approx c_d \cdot q^{1-d}.$$

Note that this is consistent with the known result of [DLP14] with $c_2 = \sqrt{e/2} \approx 1.1658$, which is also provided with heuristic analysis. Regarding this, we experimentally verify that it holds for $c_3 \approx 1.2$ as Figure 3 in Appendix C indicates.

4 IBE-Scheme from Module-NTRU

In this section, we describe our IBE scheme, whose security is based on MNTRU and Module-LWE.

4.1 Scheme Construction

We start from master key generation procedure KeyGen. It basically generates the MNTRU instance $h = (h_1, \ldots, h_{d-1}) \in R_q^{d-1}$ as the master public key and the MNTRU trapdoor matrix $T_{MNTRU} \in Z^{dn \times dn}$ as the master secret key. The master secret key elements are sampled according to Section 3.2, which implies

$$\|\tilde{T}_{MNTRU}\| = c_d \cdot q^{1-d}.$$

The detailed procedure is given by Algorithm 1.

Algorithm 1 KeyGen

Input: n, q, d
Output: $MPK = h \in R_q^{d-1}$ and $MSK = T_{MNTRU} \in Z^{dn \times dn}$

1: **for** $i = 1$ to $d-1$ **do**
2: $\quad \sigma_i \leftarrow \frac{\sqrt{d}}{\sqrt{d-i+1}} \cdot c_d \cdot q^{1-d} \cdot \sqrt{dN}$. According to Section 3.2
3: $\quad f_i \leftarrow (f_{1,i}, \ldots, f_{d,i})$ where each coefficient of $f_{j,i} \in R$ is sampled from D_{Z, σ_i}
4: **end for**
5: $S \leftarrow [f_1, \ldots, f_{d-1}]$
6: $det \leftarrow (det_1, \ldots, det_d)$ where $det_i = (-1)^{i-1} \cdot \det(S_i)$
7: $h \leftarrow det^{-1} \cdot (det_2, \ldots, det_d) \in R_q^{d-1}$
8: Find a solution $F = (F_1, \ldots, F_d) \in R^d$ of the MNTRU equation $\sum_{i=1}^{d} det_i \cdot F_i = q$
9: $T \leftarrow [A(S) \| A(F)]$
10: **return** $MPK = h$ and $MSK = T$

The extract procedure issues the user secret key sk_{id} valid for user id: The main task for this is sampling short $s \in R^d$ such that

$$\langle s, (1, h) \rangle = H(id) \mod q$$

where $H : \{0,1\}^* \to R_q$ is some hash function modeled as a random oracle. This vector s is computed by Gaussian sampling over $\tilde{T}_{MNTRU}$; and we use

GaussianSampler with the master secret key T_{MNTRU} : The standard deviation is chosen to yield KL Divergence of GaussianSampler$(T_{MNTRU}, \ )$ and the ideal discrete Gaussian $D_{p(T_{MNTRU})}$; less than $2^{\ }$: It is given by $= \eta^{0}(Z) \cdot \|T_{MNTRU}\|$ where $" = 2^{\ = 2}=(2^{\ }\overline{2}\ dn)$; and more precisely

$$\underline{c_d} \quad \sqrt{\frac{\ln 2}{2}\left(\frac{}{2} + \log_2(4^{p}\overline{2}\ dn)\right)} \quad q^{1=d}: \tag{3}$$

We also remark that this extract procedure should be stateful, i.e., it should stores every previously issued user secret keys, otherwise our scheme becomes insecure by repeated queries on the same id; actually, every IBE scheme based on GPV framework share the same feature, and some stateless variants are already argued in previous works. For simplicity we omit them and refer [GPV08]. The detailed procedure can be found in Algorithm 2 below.

Algorithm 2 Extract

Input: An identity id; the master secret key T; the master public key h and a hash function $H : f0; 1g^{} \to R_q$

Output: A user secret key $sk_{id} \in R^{d\ 1}$

1: if id is previously queried then
2: return sk_{id} in local storage
3: else
4: $t \quad (H(id); 0; \ ; 0) \in R_q^d$
5: $\underline{c_d} \quad \sqrt{\frac{\ln 2}{2}\left(\frac{}{2} + \log_2(4^{p}\overline{2}\ dn)\right)} \quad q^{1=d}$
6: $c \quad \text{GaussianSampler}(T; \ ; t)$
7: $s = (s_0; s_1; \ ; s_{d\ 1}) \quad t \quad c$. $h s; (1; h) i = t$
8: Add $sk_{id} = (s_1; \ ; s_{d\ 1})$ in local storage and return sk_{id}
9: end if

Our encryption and decryption are done in the same manner to Module-LWE based encryption. In particular, polynomials r, e_i are uniformly sampled from $f\ 1; 0; 1g^{n}$: Moreover, our IBE scheme also combines KEM and one-time-pad (OTP) as in [DLP14]. This combination of OTP is necessary for our case where the width parameter is chosen to have negligible KL divergence of Gaussian sampler; KL divergence argument only applies for search problems, and without the use of OTP, we cannot guarantee indistinguishability based security of our scheme.

For the decryption correctness, observe that

$$w = hc; (1; \ sk_{id}) i = \left\lfloor \frac{q}{2} \right\rceil m + e_0 + rs_0 \ \ \sum_{i=1}^{d\ 1} e_i s_i :$$

Then each coe cient of the error polynomial $e_0 + rs_0 \ \ \sum_{i=1}^{d\ 1} e_i s_i$ should lie over $(\ q=4; q=4)$: We rst estimate the coe cient size of the error polynomial by approximating it into (continuous) Gaussian distribution having the same

Algorithm 3 Encrypt

Input: An identity id, a message $\in \{0,1\}^m$; the master public key $h \in R_q^{d-1}$; hash functions $H : \{0,1\} \to R_q$ and $H' : \{0,1\}^n \to \{0,1\}^m$
Output: A ciphertext $C = (c, c')$ where $c \in R_q^d$ and $c' \in \{0,1\}^m$:

1: $r, e_i \leftarrow \{-1, 0, 1\}^n$ for $0 \le i \le d-1$
2: $k \leftarrow \{0,1\}^n$. k is an ephemeral key
3: $t \leftarrow H(id)$
4: $c_0 \leftarrow \langle rt + e_0 + \lfloor \frac{q}{2} \rfloor k \rangle$
5: $c_0 \leftarrow 2^{\lfloor \log_2 q \rfloor - 3} \left\langle \frac{c_0}{2^{\lfloor \log_2 q \rfloor - 3}} \right\rceil$. Store only 3 most significant bits of c_0
6: $c \leftarrow (c_0, c_1, \ldots, c_{d-1})$ where $c_i = \langle rh_i + e_i \rangle$ for $1 \le i \le d-1$
7: $c' \leftarrow H'(k)$
8: return $C = (c, c')$

Algorithm 4 Decrypt

Input: A ciphertext $C = (c, c')$; a user secret key $sk_{id} \in R^{d-1}$; and hash functions $H : \{0,1\}^* \to R_q$ and $H' : \{0,1\}^n \to \{0,1\}^m$
Output: A message $\in \{0,1\}^m$

1: $s' \leftarrow (1, sk_{id})$
2: $w \leftarrow \langle c, s' \rangle_m$
3: $k \leftarrow \lfloor \frac{2}{q} w \rceil$
4: return $m \leftarrow c \oplus H'(k)$

variance. Precisely, it is assumed to behave like 0-centered Gaussian with variance $\frac{2}{3}(\|sk_{id}\|^2 + 1)$. Using a tail bound for Gaussian distribution, we have the following condition for correctness:

$$q \ge \frac{32\sqrt{\ln 2}}{3\sqrt{3}} \|sk_{id}\|. \tag{4}$$

Moreover, as in [DLP14], one can reduce the size of ciphertext by sending only a few highest order bits of c_0, which not much harm the correctness of decryption.

4.2 Security Analysis by Attack Algorithms

In this section, we give security analysis of our IBE scheme based on the following facts from the literature. First, adapted from [PAFZ19]'s argument, if an N-dimensional lattice Λ is known to have an unusually short vector v whose size is evidently smaller than Gaussian Heuristic $\sqrt{\frac{N}{2\pi e}} \det(\Lambda)^{1/N}$, it can be found by BKZ with blocksize β satisfying

$$0.75\sqrt{N/\beta}\, \|v\| \le \delta_0^{2\beta - N} \det(\Lambda)^{1/N} \tag{5}$$

where the root Hermite factor δ_0 of BKZ β is given by $\frac{\beta}{2\pi e}(\pi\beta)^{\frac{1}{2(\beta-1)}}$ [Che13].

On the other hand, for any N-dimensional lattice Λ; if one wants to find a vector v whose size is larger than $\det(\Lambda)^{1/N}$; the required root Hermite factor δ_0 is determined by

$$\delta_0^N \approx \frac{\|v\|}{\det(\Lambda)^{1/N}}. \tag{6}$$

Based on these facts, we mount lattice attacks on several possible attack points.

Master Key Recovery. One may try to recover MSK from MPK; by finding an unusually short vector f_i in a lattice with a basis A_{MNTRU}: Since the short vector f_i is chosen to have norm smaller than $\sqrt{\frac{d}{2}}\, c_d\, q^{1/d}$; (5) implies that

$$\frac{0.75^{\frac{1}{\sqrt{dn}}}\sqrt{\frac{d}{2}}\, c_d\, q^{1/d}}{q^{1/d}} \approx 0.75^{\frac{1}{\sqrt{2n}}}\, c_d \approx \delta_0^{\sqrt{dn}}.$$

User Key Recovery. The attacker can try to obtain an user secret key id from MPK $= h$; which involves finding any short $s \in R^d$ satisfying $h s \equiv (1, a) \bmod q = H(id)$: This can be done by finding a short vector $(s, 1)$ in a $dn+1$-dimensional lattice with determinant $\sqrt{q}^n$: For correct decryption, the target vector norm would be approximately $\sqrt{dn}$ where α comes from (3). Then (6) gives a condition

$$\frac{\sqrt{dn}}{q^{n/(dn+1)}} \approx \frac{\sqrt{dn}}{q^{1/d}} \approx \delta_0^{\sqrt{dn}}.$$

IND-CPA security. Our ciphertext is of the form

$$(c_0, c_1, \ldots, c_{d-1}) = (\alpha r t + e_0, \alpha r h_1 + e_1, \ldots, \alpha r h_{d-1} + e_{d-1})$$

for MPK $= h$: Like the above user key recovery case, one can try to find the $dn+1$-dimensional vector $(e_0, \ldots, e_{d-1}, 1)$ in a lattice with determinant $\sqrt{q}^n$: Since we know the unusual short vector $(e_0, \ldots, e_{d-1}, 1)$ of size $\sqrt{2dn/3}$ in the lattice, we apply (5)

$$0.75^{\frac{1}{\sqrt{dn}}}\sqrt{2dn/3} \approx 0.75^{\frac{1}{\sqrt{dn}}}\sqrt{\frac{2}{3}}\, \delta_0^{\sqrt{dn}}\, q^{1/d}.$$

4.3 Parameter Selections

We now set a concrete parameter (d, n, q); and compare our scheme with previous results. First of all, we note that it should be noted that if one wants to use MNTRU dimension d; the master key generation involves a sampling from a discrete Gaussian with width $\alpha q^{1/d} = \alpha\sqrt{dn}$. However for $d > 3$ case, α becomes extremely small (less than 0.5) for our interest modulus q and dimension n ranges. Thus, in order

[7] One may use some portions of vectors among $u_1, \ldots, u_{d-1}$ and v; but we also have the same result.

to hedge against any possible problems regarding this extremely small discrete Gaussian, we conservatively consider only small d; explicitly $d = 3$: Moreover besides this discrete Gaussian sampling issue, too large d implies too small width parameter ; and the resulting secret matrix S would be almost zero matrix, which can be find out by simple exhaustive search.

In this regard, we instantiate our scheme with $d = 3$; with modulus parameter $q = 2^{19}$ for $n = 512$; which satisfies the correctness condition (4). Upon our security analysis of Section 4.2, the minimal block size for attacking our scheme is 506; Master key recovery requires 714; and user key recovery requires 612; and IND-CPA security requires 506: According to methodology of [ADPS16], we estimate BKZ call with block size costs $2^{0.292}$ time, and hence our instantiation provides about 147 security level. For a pair comparison, we re-evaluate security of [DLP14] parameter ($d = 2; n = 1024; q = 2^{27}$) according to our renewed security analysis of Section 4.2; Master key recovery requires 908; and user key recovery requires 867; and IND-CPA security requires 300:

Finally we also compare key sizes and ciphertext size. Clearly the ciphertext and master public key consists of d 1 elements in R_q; so their bitsizes are $(d \quad 1)n(\lfloor \log_2 q \rfloor + 1)$: Next, the user secret key consists of d 1 elements in R whose coefficients are sampled from a discrete Gaussian of standard deviation $= \frac{c_d}{\frac{\ln 2}{2} \quad \frac{}{2} + \log_2(4 \quad \frac{p}{2} \quad dn)} \quad q^{1=d}$ from (3); for our case 2:33 $q^{1=3}$, and [DLP14] case 2:28 $p \overline{q}$ (with $= 192$). This can be stored in various ways, and we follow Falcon's method that requires about $(d \quad 1)n \quad (\lfloor \log_2(\quad) \rfloor + 2)$:

We also check our proposal by a proof-of-concept implementation, and experimental results consisting speed results and concrete bit-sizes can be found in Table 1.

References

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange-a new hope. In *USENIX Security Symposium*, volume 2016, 2016.

BDK+18. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

CG17. Peter Campbell and Michael Groves. Practical post-quantum hierarchical identity-based encryption. In *16th IMA International Conference on Cryptography and Coding*, 2017. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/round-2-submissions.

Che13. Yuanmi Chen. *Réduction de réseau et sécurité concrete du chiffrement completement homomorphe*. PhD thesis, Paris 7, 2013.

DKL+18. Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.

DLP14. Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over ntru lattices. In *International Conference on the Theory and*

Application of Cryptology and Information Security, pages 22–41. Springer, 2014.

DP16. Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 191–198. ACM, 2016.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

HHGP+03. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. In *Cryptographers' Track at the RSA Conference*, pages 122–140. Springer, 2003.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

MG02. Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspectiv*. Springer, 2002.

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

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

PAFZ19. Paul Kirchner Vadim Lyubashevsky Thomas Pornin Thomas Prest Thomas Ricosset Gregor Seiler William Whyte Pierre-Alain Fouque, Jeffrey Hoffstein and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. *Post-Quantum Cryptography Standardization Round2 Submissions*, 2019. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/round-2-submissions.

PDG14. Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 353–370. Springer, 2014.

Pei16. Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

PP19. Thomas Pornin and Thomas Prest. More efficient algorithms for the ntru key generation using the field norm. In *IACR International Workshop on Public Key Cryptography*, pages 504–533. Springer, 2019.

Pre17. Thomas Prest. Sharper bounds in lattice-based cryptography using the rényi divergence. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 347–374. Springer, 2017.

SS11. Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 27–47. Springer, 2011.

A Solving generalized NTRU equation

Let $\mathsf{det} = (\mathsf{det}_1; \quad ; \mathsf{det}_d) \in R^d$ be a vector of polynomial, and let $\quad = X^n + 1$. Our goal is to find $F = (F_1; \quad ; F_d) \in R^d$ satisfying

$$\sum_{i=1} \mathsf{det}_i \quad F_i = q.$$

- First, compute $s_i \in \mathbb{Z}[X]$ such that

$$s_i \mathsf{det}_i = R_i \mod ;$$

  where $R_i \in \mathbb{Z}$ is the resultant of det_i and .
- Compute the GCD of R_i; with coefficients $u_i \in \mathbb{Z}$ such that

$$\sum_{i=1} u_i R_i = :$$

- If divides q; define

$$F_i^0 = \frac{q \quad u_i}{} s_i.$$

The vector $F^0 = (F_1^0; \quad ; F_d^0)$ may have too large size, and hence we use Babai's reduction on F^0 with a matrix S; which gives much shorter solution $F = (F_1; \quad ; F_d)$ of the MNTRU equation.

B Application to Signature

Our MNTRU trapdoor can be used for building a signature scheme. Let n be a power-of-two integer, $d \quad 2$ be a MNTRU dimension and $q > 0$ be modulus. In this case, we use the same keygen algorithm to output a public verification key $\mathsf{VK} = a$ and a secret signing key $\mathsf{SK} = \mathsf{T}_{\mathrm{MNTRU}}$. For a message ; the signing procedure runs the $\mathsf{extract}$ algorithm with $t = H(\ )$ to output a sign s. The corresponding verification procedure checks whether s is short and $hs;(1; a)i = H(\ )$. The public key size would be $(d \quad 1)n \quad d\log_2 qe$; and the signature size would be $(d \quad 1)n \quad (b\log_2(\ )c + 2)$. Falcon chooses $\quad 1{:}312 \quad k\mathsf{T} \ k$ from Rényi divergence argument due to [Pre17], which translates into $\quad 1{:}55 \quad \bar{p}\bar{q}$ in Falcon case, and $\quad 1{:}58 \quad q^{1=3}$ in our case.

For the signature usage, there is no encryption phase and we only consider the secret key recovery (the master key recovery in IBE) and the signature forgery (the user key recovery in IBE) attacks. In this case, one can check that the other attacks are only relevant to the total dimension $N = nd$; in other words, q is irrelevant to security level. Thus, under the same security level, the ring dimension n is proportional to $1{=}d$ and hence we conclude that the pk size is asymptotically proportional to $1 \quad \frac{1}{d}$; and the sign size is asymptotically proportional to $\frac{d \quad 1}{d^2}$. However, regarding the concrete parameters, such asymptotic decreases in sig size

is not so huge due to the small choice of q; and indeed the expected size of signature becomes rather larger than the NTRU case due to the constant terms. For example, Falcon chooses q to be the smallest prime such that $q = 1 \mod 2n$ (12289) for $n = 512$ and 1024 case, and $q = 1 \mod 3n$ (18433) for $n = 768$ case[8]. We focus on $n = 768$ and $d = 2$ case having total dimension 1536; where we can divide the same total dimension by $n = 512$ and $d = 3$; and use modulus $q = 12289$: Note that this two parameter sets provide the same security levels, as they have the same total dimension. The concrete sizes are compared in Table 2.

	[PAFZ19]	**Ours**
(d, n, q)	(2, 768, 18433)	(3, 512, 12289)
Bit-security	195	195
VK size (bytes)	1440	1792
Sig size (bytes)	864	892

Table 2: Comparison between [PAFZ19] and our scheme

However, we remark that our generalization can still contributes for digital signatures by introducing parameter flexibility with power-of-two dimensional rings. We leave an open question that whether many optimization techniques for power-of-two ring case are applicable, which may lead to practical (M)NTRU-based cryptosystem like MLWE-based schemes [BDK+18,DKL+18] in Post-Quantum Cryptography realm.

[8] This is for the purpose of using *number theoretic transform*(NTT), which enables fast operations on R_q.

C Plots of Section 3.2

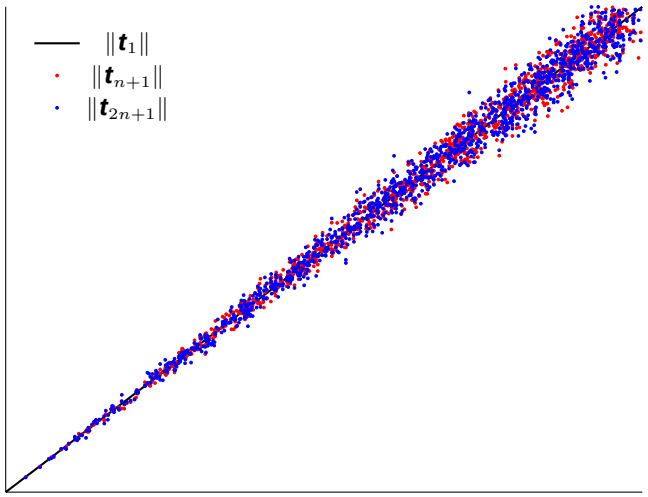

Fig. 2: $kt_{in+1}k$ values with $kt_{in+1}k = \sqrt{\frac{d}{d-i}}\ kt_1k$ for $i = 1; 2;$ with $(d; n; q) = (4; 256; 2^{27})$