Efficient Identity-Based Encryption over NTRU Lattices

Léo Ducas*, Vadim Lyubashevsky**, and Thomas Prest***

Abstract. Efficient implementations of lattice-based cryptographic schemes have been limited to only the most basic primitives like encryption and digital signatures. The main reason for this limitation is that at the core of many advanced lattice primitives is a trapdoor sampling algorithm (Gentry, Peikert, Vaikuntanathan, STOC 2008) that produced outputs that were too long for practical applications. In this work, we show that using a particular distribution over NTRU lattices can make GPV-based schemes suitable for practice. More concretely, we present the first lattice-based IBE scheme with practical parameters – key and ciphertext sizes are between two and four kilobytes, and all encryption and decryption operations take approximately one millisecond on a moderately-powered laptop. As a by-product, we also obtain digital signature schemes which are shorter than the previously most-compact ones of Ducas, Durmus, Lepoint, and Lyubashevsky from Crypto 2013.

Key words: Lattice Cryptography, Identity-Based Encryption, Digital Signatures, NTRU

1 Introduction

Recent improvements in efficiency have firmly established lattice-based cryptography as one of the leading candidates to replace number-theoretic cryptography after the eventual coming of quantum computing. There are currently lattice-based encryption [HPS98,LPR13a,LPR13b], identification [Lyu12], and digital signature schemes [GLP12,DDLL13] that have run-times (both in software and in hardware), key sizes, and output lengths that are more or less on par with traditional number-theoretic schemes. But unfortunately, the extent of practical lattice-based cryptography stops here. While number-theoretic assumptions

* University of California, San Diego. lducas@eng.ucsd.edu. This research was supported in part by the DARPA PROCEED program and NSF grant CNS-1117936. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or NSF.

** École Normale Supérieure, INRIA. vadim.lyubashevsky@inria.fr. This research was partially supported by the ANR JCJC grant "CLE".

*** École Normale Supérieure, Thales Communications & Security. thomas.prest@ens.fr.

allow for very efficient constructions of advanced schemes like identity-based encryption [BF01], group signatures [CS97,BBS04], etc. none of these schemes yet have practical lattice-based realizations.

One of the major breakthroughs in lattice cryptography was the work of Gentry, Peikert, and Vaikuntanathan [GPV08], that showed how to use a short trap-door basis to generate short lattice vectors *without revealing the trap-door*.[1] In [GPV08], this was used to give the first lattice-based construction of secure hash-and-sign digital signatures and identity-based encryption schemes. This vector-sampling algorithm has since become a key component in many other lattice constructions, ranging from hierarchical IBEs [CHKP12,ABB10] to the recent breakthrough in multi-linear map constructions [GGH13]. Unfortunately, even when using improved trap-doors [AP11,MP12] and instantiating with ideal lattices [LPR13a], signature schemes that used the GPV trap-door approach were far less practical (by about two orders of magnitude) than the Fiat-Shamir ones [GLP12,DDLL13], and identity-based encryption had ciphertexts that were even longer - having ciphertexts on the order of millions of bits.[2]

1.1 Our results

Our main result is showing that the GPV sampling procedure can in fact be used as a basis for *practical* lattice cryptography. The two main insights in our work are that one can instantiate the GPV algorithm using a particular distribution of NTRU lattices that have nearly-optimal trapdoor lengths, and that a particular parameter in the GPV algorithm can be relaxed, which results in shorter vectors being output with no loss in security. As our main applications, we propose identity-based encryption schemes that have ciphertext (and key) sizes of two and four kilobytes (for approximately 80-bit and 192-bit security, respectively) and digital signatures that have outputs (and keys) of approximately 5120 bits for about 192-bits of security. We believe that this firmly places GPV-based cryptographic schemes into the realm of practicality. The IBE outputs are orders of magnitude smaller than previous instantiations and the signature sizes are smaller by about a factor of 1.3 than in the previously shortest lattice-based scheme based on the same assumption [DDLL13].

Our schemes, like all other practical lattice-based ones, work over the polynomial ring $\mathbb{Z}_q[x]/(x^N+1)$, where N is a power of 2 and q is a prime congruent to 1 mod $2N$. For such a choice of q, the polynomial x^N+1 splits into N linear factors over $\mathbb{Z}_q$, which greatly increases the efficiency of multiplication over the ring. Our hardness assumption is related to the hardness, in the random oracle model, of solving lattice problems over NTRU lattices. These assumptions underlie the NTRU encryption scheme [HPS98], the NTRU-based fully-homomorphic encryption scheme [LTV12], and the recent signature scheme BLISS [DDLL13].

[1] A very similar algorithm was earlier proposed by Klein [Kle00], but was utilized in a different context and was not fully analyzed.

[2] The only works that we are aware of that give actual parameters for candidate constructions that use trapdoor sampling are [MP12,RS10].

Table 1. Comparing our IBE (GPV) with a recent implementation [Gui13] of the Boneh-Franklin scheme (BF). Our implementation was done in `C++`, using the `NTL` and `GnuMP` libraries. Timings were performed on an Intel Core i5-3210M laptop with a 2.5GHz CPU and 6GB RAM. The complete implementation can be found on `github.com/tprest/Lattice-IBE/`.

Scheme	**GPV-80**	**GPV-192**	**BF-128**	**BF-192**
User Secret key size	11 kbits	27 kbits	0.25 kbits	0.62 kbits
Ciphertext size	13 kbits	30 kbits	3 kbits	15 kbits
User Key Generation	**8.6 ms**	**32.7 ms**	**0.55 ms**	**3.44 ms**
Encryption	**0.91 ms**	**1.87 ms**	**7.51 ms**	**40.3 ms**
Decryption	**0.62 ms**	**1.27 ms**	**5.05 ms**	**34.2 ms**

Table 2. IBE scheme parameters (see Section 5).

Security parameter λ	**80**	**192**
Root Hermite factor [GN08] γ	1.0075	1.0044
Polynomial degree N	512	1024
Modulus q	$\approx 2^{23}$	$\approx 2^{27}$
User Public key size	13 Kbits	30 Kbits
User Secret key size	11 Kbits	27 Kbits
Ciphertext size	13 Kbits	30 Kbits
Ciphertext expansion factor	26	30

And even though this assumption is not related to the hardness of worst-case lattice problems via some worst-case to average-case reduction[3], in the fifteen years that the assumption has been around, there were no serious cryptanalytic threats against it. The work of [DDLL13] also provided experimental evidence that the computational complexity of finding short vectors in these special NTRU lattices was consistent with the extensive experiments of Gama and Nguyen on more general classes of lattices [GN08], some of which are connected to the hardness of worst-case lattice problems.

We implemented our schemes in software (see Table 1), and most of the algorithms are very efficient. The slowest one is user key generation, but this procedure is not performed often. More important is the underlying encryption scheme, which in our case is the Ring-LWE scheme from [LPR13a,LPR13b], which already has rather fast hardware implementations [PG13]. And as can be seen from the tables, decryption and encryption are very fast in software as well and compare very favorably to state-of-the-art implementations of pairing-based constructions.

[3] The work of [SS11] showed a connection between problems on NTRU lattices and worst-case problems, but for choices of parameters that do not lead to practical instantiations.

1.2 Related Work

Following the seminal work of [GPV08], there were attempts to improve several aspects of the algorithm. There were improved trap-doors [AP11], more efficient trap-door sampling algorithms [Pei10,MP12], and an NTRU signature scheme proposal [SS13]. All these papers, however, only considered parameters that preserved a security proof to lattice problems that were known to have an average-case to worst-case connection. To the best of our knowledge, our work is the first that successfully utilizes GPV trapdoor sampling in practice.

1.3 Identity-Based Encryption Scheme

In a public-key IBE scheme, the public key of every user in the system is a combination of the *master authority's* public key along with an evaluation of a publicly-computable function on the user's name or i.d.. The secret key of each user is then derived by the master authority by using his master secret key. We now give a brief description of the IBE in this paper, which is built by using the GPV algorithm to derive the user's secret keys from an NTRU lattice [GPV08,SS13], and then using the Ring-LWE encryption scheme of [LPR13a,LPR13b] for the encryption scheme.

The master public key in the scheme will be a polynomial h and the secret key will consist of a "nice basis" for the $2N$-dimensional lattice generated by the rows of $\mathbf{A}_{h,q} = \begin{pmatrix} -\mathcal{A}(h) & I_N \\ qI_N & O_N \end{pmatrix}$, where $\mathcal{A}(h)$ is the anti-circulant matrix whose i^{th} row consists of the coefficients of the polynomial $hx^i \bmod x^N + 1$ (see Definition 1). A user with identity id will have a public key consisting of h as well as $t = H(id)$, where H is some publicly-known cryptographic hash function mapping into $\mathbb{Z}_q[x]/(x^N + 1)$. The user's secret key will consist of a small polynomial s_2 such that $s_1 + s_2 h = t$, where s_1 is another small polynomial (how one generates these keys is explicited in Alg. 3 in Section 5). Encryption and decryption will proceed as in the Ring-LWE scheme of [LPR13a]. To encrypt a message $m \in \mathbb{Z}[x]/(x^N + 1)$ with binary coefficients, the sender chooses polynomials r, e_1, e_2 with small coefficients and sends the ciphertext

$$(u = rh + e_1, v = rt + e_2 + \lfloor q/2 \rfloor m).^4$$

To decrypt, the receiver computes $v - us_2 = rs_1 + e_2 + \lfloor q/2 \rfloor m - s_2 e_1$. If the parameters are properly set, then the polynomial $rs_1 + e_2 - s_2 e_1$ will have small coefficients (with respect to q), and so the coordinates in which m is 0 will be small, whereas the coordinates in which it is 1 will be close to $q/2$. Notice that for decryption, it is crucial for the polynomial $rs_1 + e_2 - s_2 e_1$ to have small coefficients, which requires s_1 and s_2 to be as small as possible.

While the above follows the usual encryption techniques based on LWE, we need a little tweak to make the security proof based on KL-divergence work

[4] In fact, one can save almost a factor of 2 in the ciphertext length by only sending the three highest order bits of v, rather than all of v

Table 3. Signature scheme parameters (see Section 5).

Security parameter λ	**80**	**192**
Root Hermite factor γ	1.0069	1.0042
Polynomial degree N	256	512
Modulus q	$\approx 2^{10}$	$\approx 2^{10}$
Public key size	2560 bits	5120 bits
Secret key size	1280 bits	2048 bits
Signature size	2560 bits	5120 bits

(see Section 4), since this argument only applies to search problems (while CPA security is a decisional problem). To do so we use a key-encapsulation mechanism, that is we encrypt a random key k rather than m, and then use it as a one-time-pad to send $m \oplus H'(k)$ where H' is a hash function.

1.4 Interlude: A Hash-and-Sign Digital Signature Scheme

The first part of the above IBE is actually a hash-and-sign digital signature scheme. The public (verification) key corresponds to the master authority's public key, the secret (signing) key is the master secret key, messages correspond to user i.d.'s, and signatures are the user secret keys. To sign a message m, the signer uses his secret key to compute short polynomials s_1, s_2 such that $s_1 + s_2 h = H(m)$, and transmits s_2. The verifier simply checks that s_2 and $H(m) - hs_2$ are small polynomials. In the IBE, the modulus q is set deliberately large to avoid decryption errors, but this is not an issue in the signature scheme. By selecting a much smaller q, which allows one to sample from a tighter distribution, the signature size can be made more compact than the user secret key size in the IBE.

In Table 3, we present some possible parameters for such signature schemes. The size of the keys and signatures compare very favorably to those of the BLISS signature scheme [DDLL13]. For example, for the 192 bit security level, the signature size in BLISS is approximately 6500 bits, whereas signatures in this paper are approximately 5000 bits. In fact, further improvements to the signature size may be possible via similar techniques that were used for BLISS. The main drawback of the hash-and-sign signature scheme is that signing requires sampling a discrete Gaussian over a *lattice*, whereas the Fiat-Shamir based BLISS scheme only required Gaussian sampling over the integers. At this point, the signature scheme in this paper yields smaller signatures but BLISS is much faster. Since both BLISS and this current proposal are very new, we believe that there are still a lot of improvements left in both constructions.

1.5 Techniques and Paper Organization

The main obstacle in making the above schemes practical is outputting short s_1, s_2 such that $s_1 + s_2 h = t$ while hiding the trap-door that allows for this generation. [GPV08] provided an algorithm where the length of s_1, s_2 crucially depend

on the length of the Gram-Schmidt orthogonalized vectors in the trap-door basis of the public lattice. In Section 3 we show, by experimental evidence backed up by a heuristic argument, that there exist distributions of NTRU lattices that have trap-doors whose lengths are within a small factor of optimal. Once we have such short trap-doors (which correspond to the master secret key in the IBE), we can use the GPV algorithm to sample s_1, s_2 such that $s_1 + s_2 h = t$. In order for (s_1, s_2) to reveal nothing about the trap-door, it's important that s_1, s_2 come from a distribution such that seeing (h, s_1, s_2, t) does not reveal whether s_1, s_2 were generated first and then t was computed as $s_1 + hs_2 = t$, or whether t was first chosen at random and then s_1, s_2 were computed using the GPV sampler.

To prove this, [GPV08] showed that the distribution of s_1, s_2 produced by their sampler is *statistically-close* to some trapdoor-independent distribution. In Section 4, we show that the requirement of statistical closeness can be relaxed, and we can instead use Kullback-Leibler divergence to obtain shorter secret keys. The intuition behind using KL-divergence can be described by the following example. If $\mathcal{B}_c$ denotes a Bernoulli variable with probability c on 1, then trying to distinguish *with constant probability* $\mathcal{B}_{1/2+\epsilon/2}$ from $\mathcal{B}_{1/2}$ requires $O(1/\epsilon^2)$ samples. Therefore if there is no adversary who can forge in time less than t (for some $t > 1/\epsilon^2$) on a signature scheme where some parameter comes from the distribution $\mathcal{B}_{1/2}$, then we can conclude that no adversary can forge in time less than approximately $1/\epsilon^2$ if that same variable were distributed according to $\mathcal{B}_{1/2+\epsilon/2}$. This is because a successful forger is also clearly a distinguisher between the two distributions (since forgeries can be checked), but no distinguisher can work in time less than $1/\epsilon^2$. On the other hand, distinguishing $\mathcal{B}_\epsilon$ from $\mathcal{B}_0$ requires only $O(1/\epsilon)$ samples. And so if there is a time t forger against a scheme using $\mathcal{B}_0$, all one can say about a forger against the scheme using $\mathcal{B}_\epsilon$ is that he cannot succeed in time less than $1/\epsilon$. In both cases, however, we have statistical distance ϵ between the two distributions. In this regard, statistical distance based arguments are not tight; but the KL-divergence is finer grained and can give tighter proofs. Indeed, in the first case, we can set $1/\epsilon$ to be the square root of our security parameter, whereas in the second case, $1/\epsilon$ would have to be the security parameter. In Section 4, we show that the GPV algorithm produces samples in a way that allows us to work with parameters for which the inverse of the statistical distance is the square root of the security parameter, whereas previous work required it to be the security parameter itself.

1.6 Conclusions and Open Problems

Trapdoor sampling is at the heart of many "advanced" lattice constructions, yet it has not been previously considered to be viable in practice. In this paper, we showed that with a proper distribution on the trap-door as well as analyzing the outputs using KL divergence instead of statistical distance, one can have schemes that are rather efficient and have their security based on the hardness of lattice problems over NTRU lattices. We believe that this opens the door to further practical implementations of lattice primitives having the GPV trap-door sampling algorithm at their core.

Our work used a distribution over NTRU lattices that is somewhat new – rather than having very short vectors, our secret key has vectors with a small Gram-Schmidt maximum. It is unclear how this compares in terms of dificulty to the hardness of lattice problems under the "standard" NTRU distribution. On the one hand, the vectors in our secret key are longer, but on the other hand, our secret key is more "orthogonal". General lattice algorithms (such as BKZ) don't seem to exploit this feature, but it is an interesting open problem whether other techniques could be used for improved cryptanalysis of our schemes.

2 Preliminaries

2.1 The Ring $\mathbb{Z}[x]/(x^N + 1)$

For the rest of the paper, N will be a power-of-two integer. We will work in the ring $\mathcal{R} \triangleq \mathbb{Z}[x]/(x^N + 1)$ (and occasionally $\mathcal{R}' \triangleq \mathbb{Q}[x]/(x^N + 1)$). Among other useful properties, $x^N + 1$ is irreducible, so $\mathcal{R}'$ is a cyclotomic field.
We clarify a few notations. 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 $\mathbb{Q}[x]$.

- fg denotes polynomial multiplication in $\mathbb{Q}[x]$, while $f*g \triangleq fg \mod (x^N+1)$.
- (f) is the vector whose coefficients are $f_0, ..., f_{N-1}$. $(f, g) \in \mathbb{R}^{2N}$ is the concatenation of (f) and (g).
- $\lfloor f \rceil$ is the coefficient-wise rounding of f. The same notation applies for vectors.

2.2 Anticirculant matrices

Definition 1 (Anticirculant matrix). *An N-dimensional anticirculant matrix of f is the following Toeplitz matrix:*

$$\mathcal{A}_N(f) = \begin{pmatrix} f_0 & f_1 & f_2 & \ddots & f_{N-1} \\ -f_{N-1} & f_0 & f_1 & \ddots & f_{N-2} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ -f_1 & -f_2 & \ddots & \ddots & f_0 \end{pmatrix} = \begin{pmatrix} (f) \\ (x * f) \\ \vdots \\ (x^{N-1} * f) \end{pmatrix}$$

When it is clear from context, we will drop the subscript N, and just write $\mathcal{A}(f)$. Anticirculant matrices verify this useful property:

Lemma 1. *Let $f, g \in \mathcal{R}$. Then $\mathcal{A}_N(f) + \mathcal{A}_N(g) = \mathcal{A}_N(f + g)$, and $\mathcal{A}_N(f) \times \mathcal{A}_N(g) = \mathcal{A}_N(f * g)$.*

2.3 Gaussian Sampling

Gaussian sampling was introduced in [GPV08] as a technique to use a short basis as a trap-door without leaking any information about the short basis; in particular it provably prevents any attack in the lines of [NR06,DN12b] designed against the NTRUSign scheme. The discrete distribution is defined as follows.

Definition 2 (Discrete Gaussians). *The n-dimensional Gaussian function $\rho_{\sigma,\boldsymbol{c}} : \mathbb{R}^n \to (0, 1]$ is defined by:*

$$\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) \stackrel{\Delta}{=} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{2\sigma^2}\right)$$

For any lattice $\Lambda \subset \mathbb{R}^n$, $\rho_{\sigma,\boldsymbol{c}}(\Lambda) \stackrel{\Delta}{=} \sum_{\boldsymbol{x}\in\Lambda} \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})$. Normalizing $\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})$ by $\rho_{\sigma,\boldsymbol{c}}(\Lambda)$, we obtain the probability mass function of the discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}$.

Using an algorithm inspired by Klein [Kle00], Gentry *et al.* [GPV08] showed that it was possible to sample vectors according to this discrete Gaussian distribution using a short basis $\mathbf{B}$ of the lattice Λ. There is a requirement on the width of the Gaussian σ related to the so called smoothing parameter. In section 4 we detail this sampler and show, using KL-divergence that the condition on the width σ can be reduced by a factor $\sqrt{2}$.

2.4 Hardness Assumptions

We can base the hardness of our IBE scheme on two assumptions that have been previously used in the literature. The first assumption deals with NTRU lattices and states that if we take two random small polynomials $f, g \in \mathcal{R}_q$, their quotient g/f is indistinguishable from random in $\mathcal{R}_q$. This assumption was first formally stated in [LTV12], but it has been studied since the introduction of the NTRU cryptosystem [HPS98] in its computational form (i.e. recovering the polynomials f and g from h). Despite more than fifteen years of cryptanalytic effort, there has not been any significant algorithmic progress towards solving either the search or decision version of this problem. As a side note, Stehle and Steinfeld [SS11] showed that for large enough f and g generated from a discrete Gaussian distribution, the quotient g/f is actually uniform in $\mathcal{R}_q$. Thus if one were to use larger polynomials, the NTRU assumption would be unnecessary. Using smaller polynomials, however, results in much more efficient schemes.

The other assumption we will be using is the Ring-LWE assumption [LPR13a] stating that the distribution of $(h_i, h_i s + e_i)$, where h_i is random in $\mathcal{R}_q$ and s, e_i are small polynomials, is indistinguishable from uniform. When the number of such samples given is polynomial (with respect to the degree of s), the coefficients of e_i cannot be too small [AG11], however, if we only give one or two samples (as is done for Ring-LWE encryption), there have been no specific attacks found if the coefficients of s, e_1, e_2 are taken from a very small set like $\{-1, 0, 1\}$. In our work, we choose to sample them from such a small set, but the scheme can be changed to sample from any other slightly larger set at the expense of slightly increasing the size of the modulus. For the concrete parameter choices, we will be using the standard methods of Gama and Nguyen [GN08] based on the currently most efficient lattice reduction algorithms [CN11].

3 Optimizing the Master Key Generation

One of the most important parameters in the scheme is the Master Secret Key: its size impacts the speed of the computations and, more importantly, the size of the users' secret keys. The smaller these secret keys will be, the more secure and efficient the scheme is (with the additional advantage that these keys can be sent more easily). While our scheme can be instantiated in any ring lattice, we choose to work in the family of NTRU lattices because the Gram-Schmidt norm of some bases are both small and easy to compute. In the end, this is what will determine the size of the users' secret keys.

3.1 The NTRU Lattices

Definition 3 (NTRU lattices). *Let N be a power-of-two integer, q a positive integer, and $f, g \in \mathcal{R}$. Let $h = g * f^{-1} \mod q$. The NTRU lattice associated to h and q is*

$$\Lambda_{h,q} = \{(u, v) \in \mathcal{R}^2 | u + v * h = 0 \mod q\}$$

$\Lambda_{h,q}$ is a full-rank lattice of $\mathbb{Z}^{2N}$ generated by the rows of $\mathbf{A}_{h,q} = \begin{pmatrix} -\mathcal{A}_N(h) & I_N \\ qI_N & O_N \end{pmatrix}$.

This basis is storage-efficient since it is uniquely defined by a single polynomial $h \in \mathcal{R}_q$, however it proves to be a poor choice if one wants to perform standard lattice operations. Assuming h is uniformly distributed in $\mathcal{R}_q$, $\mathbf{A}_{h,q}$ has a very large orthogonal defect.

This makes this basis not very appropriate for solving usual lattice problems such as finding the closest lattice vector to a target point. However, as explained in [HHGP+03], another basis can be found by computing $F, G \in \mathcal{R}$ such that:

$$f * G - g * F = q \tag{1}$$

Finding such (F, G) can be achieved efficiently and we describe one way (which is not new) of doing it in Section 5. A short basis is then provided by the following proposition.

Proposition 1. *Let $f, g, F, G \in \mathcal{R}$ verifying (1) and $h = g * f^{-1} \mod q$. Then $\mathbf{A}_{h,q} = \begin{pmatrix} -\mathcal{A}(h) & I_N \\ qI_N & O_N \end{pmatrix}$ and $\mathbf{B}_{f,g} = \begin{pmatrix} \mathcal{A}(g) & -\mathcal{A}(f) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$ generate the same lattice.*

Proof. Consider $\mathbf{P} = \mathbf{A}_{h,q} \times \mathbf{B}_{f,g}^{-1}$ the change-of-basis matrix between $\mathbf{A}_{h,q}$ and $\mathbf{B}_{f,g}$. One can check that $q\mathbf{P} = O_{2N} \mod q$, so $\mathbf{P} \in \mathbb{Z}^{2N \times 2N}$. Also, $|\det(\mathbf{P})| = 1$ so $\mathbf{P}^{-1} \in \mathbb{Z}^{2N \times 2N}$. We can conclude that $\mathbf{A}_{h,q}$ and $\mathbf{B}_{f,g}$ both generate the same lattice. □

Definition 4 (Gram-Schmidt norm [GPV08]). *Let $\mathbf{B} = (\mathbf{b}_i)_{i \in I}$ be a finite basis, and $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_i)_{i \in I}$ be its Gram-Schmidt orthogonalization. The Gram-Schmidt norm of $\mathbf{B}$ is the value*

$$\|\tilde{\mathbf{B}}\| = \max_{i \in I} \|\tilde{\mathbf{b}}_i\|$$

An interesting property of NTRU lattices is related to the Gram-Schmidt norm of their bases: they can be small and can be computed quickly. These two facts and their benefits for our scheme are discussed in the following subsection.

3.2 Bounding the Gram-Schmidt norm

The lattice over which we do Gaussian sampling is $\Lambda_{h,q}$, and the size of the secret keys we sample will be proportional to $\|\tilde{\mathbf{B}}\|$, where $\mathbf{B}$ is the basis of $\Lambda_{h,q}$ used in the Gaussian sampler. It is very important then that the Gram-Schmidt norm $\|\tilde{\mathbf{B}}\|$ of $\mathbf{B}$ is as small as possible.
Proposition 1 tells us that $\mathbf{B}_{f,g}$ is a basis of $\Lambda_{h,q}$. The second step is to compute its Gram-Schmidt norm $\|\tilde{\mathbf{B}}_{f,g}\|$. For general lattices, this is done by applying the Gram-Schmidt process to the basis and computing the maximum length of the resulting vectors. In the case of NTRU lattices, however, Lemmas 2 and 3 allow to compute $\|\tilde{\mathbf{B}}_{f,g}\|$ much faster, in time $\mathcal{O}(N \log(N))$ instead of $\mathcal{O}(N^3)$.

Lemma 2. *Let $\mathbf{B}_{f,g}$ be as defined in Proposition 1, and $\mathbf{b}_1, ..., \mathbf{b}_{2N}$ be the row vectors of $\mathbf{B}_{f,g}$. Then $\|\tilde{\mathbf{B}}_{f,g}\| = max\{\|\tilde{\mathbf{b}}_1\|, \|\tilde{\mathbf{b}}_{N+1}\|\}$*

Proof. For $\mathbf{V}$ a subspace of $\mathbb{R}^{2N}$ and $\mathbf{b} \in \mathbb{R}^{2N}$, let us denote $\mathbf{Proj}_{\mathbf{V}}(\mathbf{b})$ the orthogonal projection of $\mathbf{b}$ over $\mathbf{V}$. We also call r the linear isometry $(f, g) \mapsto (x * f, x * g)$ (see the notations from Subsection 2.1), so that for any $i \leqslant N$, $\mathbf{b}_i = r^{i-1}(\mathbf{b}_1)$ and $\mathbf{b}_{N+i} = r^{i-1}(\mathbf{b}_{N+1})$. Let $\mathbf{V}_i = Span(\mathbf{b}_1, ..., \mathbf{b}_i)^{\perp}$. By definition of the Gram-Schmidt orthogonalization, for any $i \in [\![1, 2N]\!]$, $\tilde{\mathbf{b}}_i = \mathbf{Proj}_{\mathbf{V}_{i-1}}(\mathbf{b}_i)$.
Moreover, one can check the two following properties :

- $\|\mathbf{Proj}_{\mathbf{V}}(\mathbf{b})\| \leqslant \|\mathbf{b}\|$
- $\mathbf{V} \subseteq \mathbf{W} \Rightarrow \|\mathbf{Proj}_{\mathbf{V}}(\mathbf{b})\| \leqslant \|\mathbf{Proj}_{\mathbf{W}}(\mathbf{b})\|$

From the first property comes the fact that for any $i \in [\![1, N]\!]$, $\|\tilde{\mathbf{b}}_i\| \leqslant \|\mathbf{b}_1\| = \|\tilde{\mathbf{b}}_1\|$. Proving $\|\tilde{\mathbf{b}}_{N+i}\| \leqslant \|\tilde{\mathbf{b}}_{N+1}\|$ is a bit trickier. Since $Span(\mathbf{b}_1, ...\mathbf{b}_N)$ is stable by r, so is $\mathbf{V}_N$. One can check that $\mathbf{Proj}_{\mathbf{V}_N}(\mathbf{b}_{N+i}) = r^{i-1}(\mathbf{Proj}_{\mathbf{V}_N}(\mathbf{b}_{N+1}))$. Now $\mathbf{V}_{N+i-1} \subseteq \mathbf{V}_N$, so :
$\|\tilde{\mathbf{b}}_{N+i}\| = \|\mathbf{Proj}_{\mathbf{V}_{N+i-1}}(\mathbf{b}_{N+i})\| \leqslant \|\mathbf{Proj}_{\mathbf{V}_N}(\mathbf{b}_{N+i})\| = \|\mathbf{Proj}_{\mathbf{V}_N}(\mathbf{b}_{N+1})\| = \|\tilde{\mathbf{b}}_{N+1}\|$
Which concludes this proof. □

Figure 1 illustrates the result of Lemma 2. Before the reduction, all the vectors of each semi-basis are of the same size, but after the reduction, the largest vector of $\tilde{\mathbf{B}}_{f,g}$ is either $\tilde{\mathbf{b}}_1$ or $\tilde{\mathbf{b}}_{N+1}$.

Instead of computing $2N$ values $\|\tilde{\mathbf{b}}_1\|, ..., \|\tilde{\mathbf{b}}_{2N}\|$, there is now only two of them to compute. We already know that $\|\tilde{\mathbf{b}}_1\| = \|\mathbf{b}_1\|$, and we introduce a notation which will provide us an expression for $\|\tilde{\mathbf{b}}_{N+1}\|$.

Definition 5. *Let $f \in \mathcal{R}'$. We denote $\bar{f}$ the unique polynomial in $\mathcal{R}'$ such that $\mathcal{A}(f)^t = \mathcal{A}(\bar{f})$. If $f(x) = \sum_{i=0}^{N-1} f_i x^i$, then $\bar{f}(x) = f_0 - \sum_{i=1}^{N-1} f_{N-i} x^i$*

This notation is only needed in the master key generation, to compute the Gram-Schmidt norm of the basis as well as reducing $(G, -F)$ modulo $(g, -f)$. The following lemma gives an exact expression for $\|\tilde{\mathbf{b}}_{N+1}\|$.

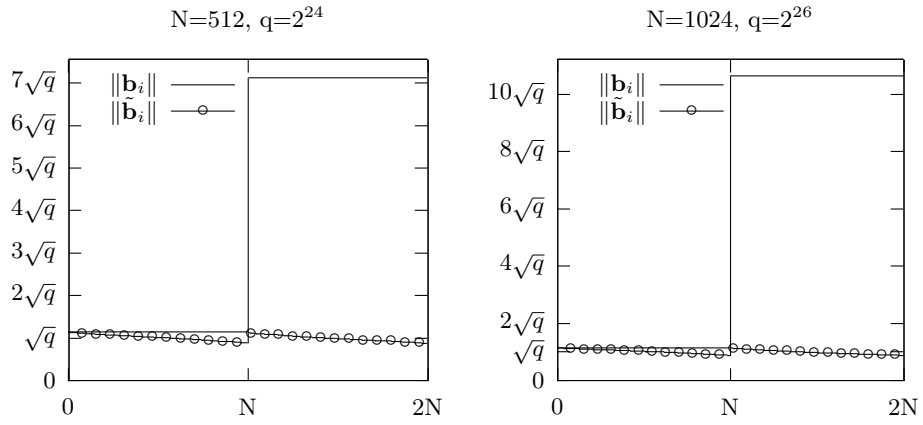


Fig. 1. Size of the vectors of $\tilde{\mathbf{B}}_{f,g}$ before and after Gram-Schmidt reduction

Lemma 3. $\|\tilde{\mathbf{b}}_{N+1}\| = \left\|\left(\frac{q\bar{f}}{f*\bar{f}+g*\bar{g}}, \frac{q\bar{g}}{f*\bar{f}+g*\bar{g}}\right)\right\|$

Proof. Let $k = \frac{\bar{f}*F+\bar{g}*G}{f*\bar{f}+g*\bar{g}}$ *mod* (x^N+1) and write $k(x) = \sum_{i=0}^{N-1} k_i x^i$. Then

$$c \stackrel{\Delta}{=} \mathbf{b}_{N+1} - \sum_{i=0}^{N-1} k_i \mathbf{b}_{i+1} = (G,-F) - (k*g,-k*f) = \left(\frac{q\bar{f}}{f*\bar{f}+g*\bar{g}}, \frac{q\bar{g}}{f*\bar{f}+g*\bar{g}}\right)$$

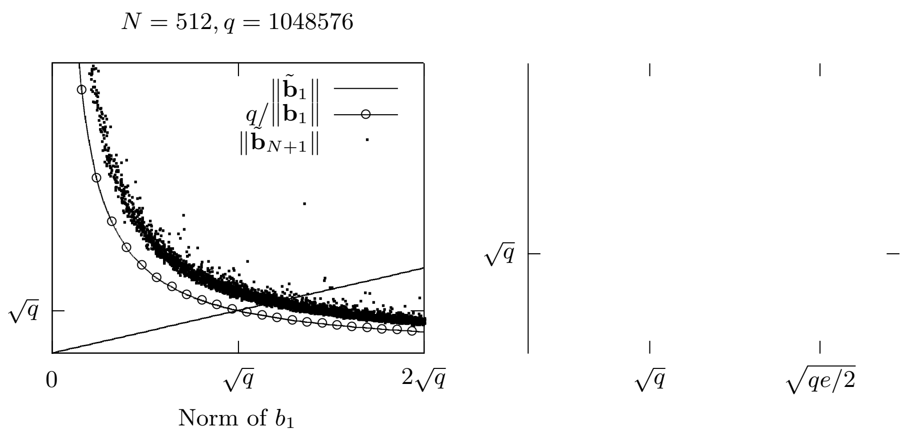
is orthogonal to $Span(\mathbf{b}_1, ..., \mathbf{b}_N)$. By the uniqueness of the Gram-Schmidt decomposition, $c = \tilde{\mathbf{b}}_{N+1}$ and the result follows. □

This enables us to compute $\|\tilde{\mathbf{B}}_{f,g}\|$ only from (f,g), gaining some time when generating the Master Key. Knowing (f,g) is enough to know almost instantly whether the basis $\tilde{\mathbf{B}}_{f,g}$ will be a good one for Gaussian sampling. After deriving this formula for $\|\tilde{\mathbf{B}}_{f,g}\|$, we ran experiments to compute it for different values of N, q and initial vector $\mathbf{b}_1$.

For fixed values of $\|\mathbf{b}_1\|$ and q, experiments show no correlation between the dimension N and $\|\tilde{\mathbf{B}}_{f,g}\|$. Moreover, they suggest that $\|\tilde{\mathbf{b}}_{N+1}\|$ is actually pretty close to its lower bound $q/\|\mathbf{b}_1\|$ (see Lemma 4 for the proof of this bound). Both experiments and a heuristic indicate that the optimal choice for $\|\mathbf{b}_1\|$ is $\|\mathbf{b}_1\| \approx \sqrt{\frac{qe}{2}}$, since we then get $\|\tilde{\mathbf{b}}_{N+1}\| \approx \|\tilde{\mathbf{b}}_1\|$. And so in our experiments, we sample $\mathbf{b}_1$ with a norm slightly bigger than $\sqrt{\frac{qe}{2}} \approx 1.1658\sqrt{q}$. The heuristic can be found in the full version of this paper.

The following lemma provides a theoretical lower bound for $\|\tilde{\mathbf{b}}_{N+1}\|$, given q and $\|\mathbf{b}_1\|$. In our case, $\|\tilde{\mathbf{b}}_{N+1}\|$ is very close to its lover bound.

Lemma 4. *Let* $\mathbf{B} = (\mathbf{b}_i)_{1 \leqslant i \leqslant 2N}$ *be a NTRU basis.* $\|\tilde{\mathbf{b}}_{N+1}\|$ *admits the following lower bound:* $\|\tilde{\mathbf{b}}_{N+1}\| \geqslant q/\|\mathbf{b}_1\|$.



1

sufficient conditiontoensure

Algorithm 1 `Gaussian_Sampler`$(\mathbf{B}, \sigma, \mathbf{c})$

Require: Basis $\mathbf{B}$ of a n-dimensional lattice Λ, standard deviation $\sigma > 0$, center $\mathbf{c} \in \mathbb{Z}^n$
Ensure: $\mathbf{v}$ sampled in $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$
1: $\mathbf{v}_n \leftarrow \mathbf{0}$
2: $\mathbf{c}_n \leftarrow \mathbf{c}$
3: **for** $i \leftarrow n, ..., 1$ **do**
4: $\quad c'_i \leftarrow \langle \mathbf{c}_i, \tilde{\mathbf{b}}_i \rangle / \|\tilde{\mathbf{b}}_i\|^2$
5: $\quad \sigma'_i \leftarrow \sigma / \|\tilde{\mathbf{b}}_i\|$
6: $\quad z_i \leftarrow$ `SampleZ`(σ'_i, c'_i)
7: $\quad \mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$ and $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$
8: **end for**
9: **return** $\mathbf{v}_0$

The sub-algorithm `SampleZ`(σ', c') samples a 1-dimensional Gaussian $\mathcal{D}_{\mathbb{Z},\sigma',c'}$. This can be achieved in various ways: rejection sampling, look-up tables, etc. For our implementation, we chose an hybrid method using the discrete Gaussian sampling from [DDLL13] and "standard" rejection sampling.

Fig. 3. Description of Klein-GPV Gaussian Sampler

In this section, we sketch why the condition $\epsilon = 2^{-\lambda}/(4N)$ can be relaxed to

$$\epsilon \leq 2^{-\lambda/2}/(4\sqrt{2}N);$$

asymptotically square-rooting the minimal value of ϵ; this impacts the value of $\eta'_\epsilon(\mathbb{Z})$ by a factor $\sqrt{2}$, that is one can use *the same algorithm* with a standard deviation σ shorter by a factor $\sqrt{2}$. To do so we rely on a finer grained measure of "distance"[5] between distributions, called Kullback-Leibler Divergence (or KL Divergence). Interestingly, this common notion from information theory, has to our knowledge been used in cryptography only in the context of symmetric key cryptanalysis [Vau03]. The use of KL Divergence recently found similar application in lattice based Cryptography, namely for an enhanced implementation [PDG14] of Bliss [DDLL13]. It is defined as follows:

Definition 7 (Kullback-Leibler Divergence). *Let $\mathcal{P}$ and $\mathcal{Q}$ be two distributions over a common countable set Ω, and let $S \subset \Omega$ be the support of $\mathcal{P}$. The Kullback-Leibler Divergence, noted D_{KL} of $\mathcal{Q}$ from $\mathcal{P}$ is defined as:*

$$D_{KL}(\mathcal{P} \| \mathcal{Q}) = \sum_{i \in S} \ln \left(\frac{\mathcal{P}(i)}{\mathcal{Q}(i)} \right) \mathcal{P}(i)$$

with the convention that $\ln(x/0) = +\infty$ for any $x > 0$.

[5] Technically it is not a distance: it is neither symmetric nor does it verifies the triangle inequality.

For complements the reader can refer to [CT91]. We only require two essential properties : additivity: $D_{\mathrm{KL}}(\mathcal{P}_0 \times \mathcal{P}_1 \| \mathcal{Q}_0 \times \mathcal{Q}_1) = D_{\mathrm{KL}}(\mathcal{P}_0 \| \mathcal{Q}_0) + D_{\mathrm{KL}}(\mathcal{P}_1 \| \mathcal{Q}_1)$, and the inequality $D_{\mathrm{KL}}(f(\mathcal{P}) \| f(\mathcal{Q})) \leq D_{\mathrm{KL}}(\mathcal{P} \| \mathcal{Q}))$.

Lemma 5 (Bounding Success Probability Variations [PDG14]). *Let $\mathcal{E}^{\mathcal{P}}$ be an algorithm making at most q queries to an oracle sampling from a distribution $\mathcal{P}$ and outputting a bit. Let $\epsilon \geq 0$, $\mathcal{Q}$ be a distribution such that $D_{KL}(\mathcal{P} \| \mathcal{Q}) \leq \epsilon$, and x (resp. y) denote the probability that $\mathcal{E}^{\mathcal{P}}$ (resp. $\mathcal{E}^{\mathcal{Q}}$) outputs 1. Then,*

$$|x - y| \leq \frac{1}{\sqrt{2}} \sqrt{q\epsilon}.$$

Concrete security. This lemma lets us conclude that if a scheme is λ-bit secure with access to a perfect oracle for distribution $\mathcal{P}$, then it is also about λ-bit secure with oracle access to $\mathcal{Q}$ if $D_{\mathrm{KL}}(\mathcal{P} \| \mathcal{Q}) \leq 2^{-\lambda}$.

To argue concrete security according to Lemma 5, consider a search problem $\mathcal{S}^{\mathcal{P}}$ using oracle access to a distribution $\mathcal{P}$, and assume it is not λ-bit hard; that is there exists an attacker $\mathcal{A}$ that solve $\mathcal{S}^{\mathcal{P}}$ with probability p and has running time less than $2^{\lambda}/p$; equivalently (repeating the attack until success) there exists an algorithm $\mathcal{A}'$ that solves $\mathcal{S}^{\mathcal{P}}$ in time $\approx 2^{\lambda}$ with probability at least $3/4$. Such algorithms make $q \leq 2^{\lambda}$ queries to $\mathcal{P}$. If $D_{\mathrm{KL}}(\mathcal{P} \| \mathcal{Q}) \leq 2^{-\lambda}$, Lemma 5 ensures us that the success of $\mathcal{A}'$ against $\mathcal{S}^{\mathcal{Q}}$ will be at least $1/4$; in other word if $\mathcal{S}^{\mathcal{Q}}$ is λ-bit secure, $\mathcal{S}^{\mathcal{P}}$ is also about λ-bit secure.

Note that this applies to search problems only, therefore, it is unclear if it could be directly applied to any CPA scheme: CPA security is a decisional problem, not a search problem. Yet our IBE design makes this argument valid: we designed encryption using key-encapsulation mechanism, the random key k being fed into a hash function H' to one-time-pad the message. Modeling H' as a random oracle, one easily proves that breaking CPA security with advantage p is as hard as recovering k with probability p; which is a search problem. □

The point is that the KL Divergence is in some cases much smaller than statistical distance; and it will indeed be the case for Klein sampling as used in [GPV08] and described in Fig. 3.

Theorem 2 (KL Divergence of the Gaussian Sampler). *For any $\epsilon \in (0, 1/4n)$, if $\sigma \geqslant \eta'_{\epsilon}(\mathbb{Z}) \cdot \|\tilde{\mathbf{B}}\|$ then the KL Divergence between $D_{\Lambda(\mathbf{B}),\mathbf{c},\sigma}$ and the output of* $\texttt{Gaussian\_Sampler}(\mathbf{B}, \sigma, \mathbf{c})$ *is bounded by* $2\left(1 - \left(\frac{1+\epsilon}{1-\epsilon}\right)^n\right)^2 \approx 8n^2\epsilon^2$.

Proof. The probability that Klein's algorithm outputs $\mathbf{x} = \tilde{\mathbf{x}}$ on inputs $\sigma, \mathbf{B}, \mathbf{c}$ is proportional to

$$\prod_{i=1}^{n} \frac{1}{\rho_{\sigma_i, c'_i}(\mathbb{Z})} \cdot \rho_{\sigma,\mathbf{c}}(\tilde{\mathbf{x}})$$

for $\sigma_i = \sigma / \|\tilde{\mathbf{b}}_i\|$ and some $c'_i \in \mathbb{R}$ that depends on $\mathbf{c}$ and $\mathbf{B}$. as detailed in [GPV08]. By assumption, $\sigma_i \geq \eta_{\epsilon}(\mathbb{Z})$, therefore $\rho_{\sigma_i, c'_i}(\mathbb{Z}) \in [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_{\sigma_i}(\mathbb{Z})$

(see [MR07, Lemma 4.4]). The relative error to the desired distribution (proportional to $\rho_{\sigma,\mathbf{c}}(\tilde{\mathbf{x}})$) is therefore bounded by $1 - \left(\frac{1+\epsilon}{1-\epsilon}\right)^n$; we can conclude using Lemma 2 from [PDG14]. □

This last theorem implies that the condition $\epsilon \leq \frac{2^{-\lambda}}{4N}$ of [DN12a] can be relaxed to $\epsilon \leq \frac{2^{-\lambda/2}}{4\sqrt{2}N}$ which conclude this section.

5 The Schemes and their Security Analysis

5.1 The IBE Scheme

We recall that an IBE scheme is composed of four algorithms: `Master_Keygen`, which generates the Master Keys, `Extract`, which uses the Master Secret Key to generate users' secret keys for any identity, `Encrypt`, which allows anybody to encrypt a message for an user given the Master Public Key and the user's identity, and `Decrypt` which enables an user to decrypt the messages intended to him with his secret key.

Algorithm 2 $\mathsf{Master\_Keygen}(N, q)$

Require: N, q
Ensure: Master Secret Key $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$ and Master Public Key $h \in \mathcal{R}_q$
1: $\sigma_f = 1.17\sqrt{\frac{q}{2N}}$ {σ_f chosen such that $\mathbb{E}[\|b_1\|] = 1.17\sqrt{q}$}
2: $f, g \leftarrow \mathcal{D}_{N,\sigma_f}$
3: Norm $\leftarrow \max\left(\|(g,-f)\|, \left\|\left(\frac{g\bar{f}}{f*\bar{f}+g*\bar{g}}, \frac{g\bar{g}}{f*\bar{f}+g*\bar{g}}\right)\right\|\right)$ {We compute $\|\tilde{\mathbf{B}}_{f,g}\|$}
4: if Norm$> 1.17\sqrt{q}$, go to step 2
5: Using extended euclidean algorithm, compute $\rho_f, \rho_g \in \mathcal{R}$ and $R_f, R_g \in \mathbb{Z}$ such that
   - $\rho_f \cdot f = R_f \mod (x^N + 1)$
   - $\rho_g \cdot g = R_g \mod (x^N + 1)$
6: if $GCD(R_f, R_g) \neq 1$ or $GCD(R_f, q) \neq 1$, go to step 2
7: Using extended euclidean algorithm, compute $u, v \in \mathbb{Z}$ such that $u \cdot R_f + v \cdot R_g = 1$
8: $F \leftarrow qv\rho_g, G \leftarrow -qu\rho_f$
9: $k = \left\lfloor \frac{F*\bar{f}+G*\bar{g}}{f*\bar{f}+g*\bar{g}} \right\rceil \in \mathcal{R}$
10: Reduce F and G: $F \leftarrow F - k*f, G \leftarrow G - k*g$
11: $h = g * f^{-1} \mod q$
12: $\mathbf{B} = \begin{pmatrix} \mathcal{A}(g) & -\mathcal{A}(f) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{pmatrix}$

Here, $\mathbf{B}$ is a short basis of $\Lambda_{h,q}$, making it a trapdoor for sampling short elements (s_1, s_2) such that $s_1 + s_2 * h = t$ for any t, without leaking any information about itself.

Algorithm 3 stores the secret key for each user that has made a query. The reasons behind this choice, and alternatives are discussed in the full version of this paper.

Algorithm 3 $\texttt{Extract}(\mathbf{B}, id)$

Require: Master Secret Key $\mathbf{B} \in \mathbb{Z}_q^{2N \times 2N}$, hash function $H : \{0,1\}^* \to \mathbb{Z}_q^N$, user identity id
Ensure: User secret key $\mathbf{SK}_{id} \in \mathcal{R}_q$
1: **if** $\mathbf{SK}_{id}$ is in local storage **then**
2: Output $\mathbf{SK}_{id}$ to user id
3: **else**
4: $t \leftarrow H(id) \in \mathbb{Z}_q^N$
5: $(s_1, s_2) \leftarrow (t, 0) - \texttt{Gaussian\_Sampler}(\mathbf{B}, \sigma, (t, 0))$ $\{s_1 + s_2 * h = t\}$
6: $\mathbf{SK}_{id} \leftarrow s_2$
7: Output $\mathbf{SK}_{id}$ to user id and keep it in local storage
8: **end if**

Algorithm 4 $\texttt{Encrypt}(id, m)$

Require: Hash functions $H : \{0,1\}^* \to \mathbb{Z}_q^N$ and $H' : \{0,1\}^N \to \{0,1\}^m$, message $m \in \{0,1\}^m$, Master Public Key $h \in \mathcal{R}_q$, identity id
Ensure: Encryption $(u, v, c) \in \mathcal{R}_q^2$ of m under the public key of id
1: $r, e_1, e_2 \leftarrow \{-1, 0, 1\}^N$; $k \leftarrow \{0,1\}^N$ (uniform)
2: $t \leftarrow H(id)$
3: $u \leftarrow r * h + e_1 \in \mathcal{R}_q$
4: $v \leftarrow r * t + e_2 + \lfloor q/2 \rfloor \cdot k \in \mathcal{R}_q$
5: Drop the least significant bits of v: $v \leftarrow 2^\ell \left\lfloor v/2^\ell \right\rceil$
6: Output $(u, v, m \oplus H'(k))$

Note that encryption is designed using a key-encapsulation mechanism; the hash of the key k is used to one-time-pad the message. If H' is modeled as a random oracle, this makes the CPA security (a decisional problem) of the scheme as hard as finding the key k exactly (a search problem). Basing the security argument on a search problem is necessary for our KL Divergence-based security argument to hold, as explained in Section 4.

Algorithm 5 $\texttt{Decrypt}(\mathbf{MS}_{id}, (u, v, c))$

Require: User secret key $\mathbf{SK}_{id}$, encryption (u, v, c) of m
Ensure: Message $m \in \{0,1\}^N$
1: $w \leftarrow v - u * s_2$
2: $k \leftarrow \left\lfloor \frac{w}{q/2} \right\rceil$
3: Output $m \leftarrow c \oplus H'(k)$

In order for an user to decrypt correctly, $y = r * s_1 + e_2 - e_1 * s_2$ must have all its coefficients in $(-\frac{q}{4}, \frac{q}{4})$, so we need to set q big enough. In practice, this gives $q \geqslant 5.1 \cdot 10^6$ for $\lambda = 80$, and $q \geqslant 5.6 \cdot 10^6$ for $\lambda = 192$.

Dropping bits can also lead to incorrect decryption. However, for $\ell \leqslant \lfloor \log_2 q \rfloor - 3$, it doesn't significantly affect the correct decryption rate of the scheme, so we take this value of ℓ as standard.

The computations leading to these values of q and ℓ can be found in the full version of this paper.

The scheme described above is only CPA-secure. In practice, we would want to make it CCA-secure by using one of the standard transformations (e.g. [FO99]).

5.2 Security analysis of the IBE scheme

We now use the techniques in [GN08,CN11,DDLL13] to analyze the concrete security of the scheme. The way lattice schemes are analyzed is to determine the hardness of the underlying lattice problem, which is measured using the "root Hermite factor" introduced in [GN08]. If one is looking for a vector $\mathbf{v}$ in an n-dimensional lattice that is *larger* than the n^{th} root of the determinant, then the associated root Hermite factor is

$$\frac{\|\mathbf{v}\|}{det(\Lambda)^{1/n}} = \gamma^n \tag{2}$$

If one is looking for an unusually-short planted vector $\mathbf{v}$ in an NTRU lattice, then the associated root Hermite factor, according to the experiments in [DDLL13] is

$$\frac{\sqrt{n/(2\pi e)} \cdot det(\Lambda)^{1/n}}{\|\mathbf{v}\|} = .4\gamma^n. \tag{3}$$

Based on the results in [GN08,CN11], one can get a very rough estimate of the hardness of the lattice problem based on the value of γ (unfortunately, there has not been enough lattice cryptanalysis literature to have anything more that just a rough estimate). For values of $\gamma \approx 1.007$, finding the vector is at least 2^{80}-hard. For values less that 1.004, the problem seems completely intractable and is approximated to be at least 192-bits hard.

The most vulnerable part of our IBE scheme will be the actual encryption. Still, we will first run through the best attacks on the master public key and user secret keys because these correspond exactly to attacks on the key and signature forgery, respectively, in the hahs-and-sign digital signature scheme. Our master public key polynomial h is not generated uniformly at random, but rather as $g * f^{-1}$. The best-known attack for distinguishing an NTRU polynomial from a random one is to find the polynomials f, g that are "abnormally short". This involves finding the short f and g such that $h * f - g = 0 \bmod q$. This is equivalent to finding the vector (f, g) in a $2N$-dimensional lattice with determinant q^N. From Section 3, we know that the euclidean norm of the vector (f, g) is approximately $1.17\sqrt{q}$ and so calculating the value of γ using (3), we get

$$\frac{\sqrt{2N/(2\pi e)} \cdot \sqrt{q}}{1.17\sqrt{q}} = .4\gamma^{2N} \Longrightarrow \gamma = (\sqrt{N}/1.368)^{1/2N},$$

which is 1.0054 for $N = 256$ and 1.0027 for $N = 512$, which is already beyond the realm of practical algorithms. The secret user keys (s_1, s_2) are generated

with standard deviation of about $\sigma = 1.17\eta'_\epsilon(\mathbb{Z}) \cdot \|\tilde{\mathbf{B}}\|$, which gives $\sigma = 1.5110\sqrt{q}$ for $N = 256$ (resp. $\sigma = 2.2358\sqrt{q}$ for $N = 512$), and so the vector has length $\sigma\sqrt{2N}$, which by (2) results in a value of γ,

$$\frac{\sigma\sqrt{2N}}{\sqrt{q}} = \gamma^{2N} \Longrightarrow \begin{cases} \gamma = (2.137\sqrt{N})^{1/2N} \text{ for } N = 256 \\ \gamma = (3.162\sqrt{N})^{1/2N} \text{ for } N = 512 \end{cases}$$

which is 1.0069 for $N = 256$ and 1.0042 for $N = 512$.

We now move on to the hardness of breaking the CPA-security of the scheme. Encryption (disregarding the message) consists of $(u = r * h + e_1, v = r * t + e_2)$, where the coefficients of r, e_1, e_2 have coefficients chosen from $\{-1, 0, 1\}$. In order to avoid decryption errors, the value of the modulus q has to be set fairly high (see Table 1). The best-known attack against the encryption scheme involves essentially recovering the errors e_1, e_2. From the ciphertext (u, v), we can set up the equation $(t * h^{-1}) * e_1 - e_2 = (t * h^{-1}) * u - v \bmod q$, which can be converted into the problem of finding the $2N + 1$-dimensional vector $(e_1, e_2, 1)$ in a $2N + 1$-dimensional lattice with determinant q^N. Using (3), we get

$$\frac{\sqrt{2N/(2\pi e)} \cdot \sqrt{q}}{\|(e_1, e_2, 1)\|} = .4\gamma^{2N} \Longrightarrow \gamma = (.74\sqrt{q})^{1/2N},$$

which gives us $\gamma = 1.0075$ for $N = 512$ and $\gamma = 1.0044$ for $N = 1024$.

5.3 Analysis of the signature scheme

In our signature scheme, the Keygen is provided by Algorithm 2, Signature by Algorithm 3 and Verification by checking the norm of (s_1, s_2) as well as the equality $s_1 + s_2 * h = H(message)$. Since there is no encryption, we can discard the CPA-security analysis at the end of the previous section, as well as the issues regarding correctness of the encryption. This leads to much smaller values for N and q, which can be found in the Table 3 of Section 1.

We now analyze the bitsize of the secret key, public key and signature. The public key is $h \in \mathcal{R}_q$, as well as the signature s_1, so their bitsizes are $N\lceil \log_2 q \rceil$. The secret key is f such that $f * h = g \mod q$. Given the procedure to generate $\mathbf{b}_1 = (f, g)$, with high probability each coefficient of f has absolute value at most equal to $6\sigma_f$ (if it isn't the case, one just need to resample the coefficient). f can therefore be stored using $N(1 + \lceil \log_2(6\sigma_f) \rceil)$ bits, where $\sigma_f = 1.17\sqrt{\frac{q}{2N}}$.[6]

Acknowledgments

The authors wish to thank David Xiao and Aurore Guillevic for helpful conversations, as well as the anonymous Asiacrypt'14 reviewers.

[6] Using Huffman coding, as in BLISS [DDLL13], may also be appropriate here for reducing the key length by several hundred bits

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *CRYPTO*, pages 98–115, 2010.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415, 2011.

[AP11] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. CRYPTO '01, pages 213–229, 2001.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[CN11] Yuanmi Chen and Phong Q. Nguyen. Bkz 2.0: better lattice security estimates. ASIACRYPT'11, pages 1–20, 2011.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO*, pages 410–424, 1997.

[CT91] Thomas M. Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 1991.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians - crypto 2013. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*. Springer, 2013.

[DN12a] Léo Ducas and Phong Q. Nguyen. Faster gaussian lattice sampling using lazy floating-point arithmetic. ASIACRYPT'12, pages 415–432, Berlin, Heidelberg, 2012. Springer-Verlag.

[DN12b] Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more: cryptanalysis of ntrusign countermeasures. ASIACRYPT'12, pages 433–450, Berlin, Heidelberg, 2012. Springer-Verlag.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, pages 537–554, 1999.

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, pages 530–547, 2012.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. EUROCRYPT'08, pages 31–51, Berlin, Heidelberg, 2008. Springer-Verlag.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[Gui13] Aurore Guillevic. *Étude de l'arithmétique des couplages sur les courbes algébriques pour la cryptographie*. These, Ecole Normale Supérieure de Paris - ENS Paris, December 2013.

[HHGP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. Ntrusign: digital signatures using the ntru lattice. CT-RSA'03, pages 122–140, 2003.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. ANTS-III, pages 267–288, 1998.

[Kle00] Philip Klein. Finding the closest lattice vector when it's unusually close. SODA '00, pages 937–941, 2000.

[LPR13a] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43, 2013. Preliminary version appeared in EUROCRYPT 2010.

[LPR13b] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In *EUROCRYPT*, pages 35–54, 2013.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234, 2012.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.

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

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

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: cryptanalysis of ggh and ntru signatures. EUROCRYPT'06, pages 271–288, Berlin, Heidelberg, 2006. Springer-Verlag.

[PDG14] Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. *IACR Cryptology ePrint Archive*, 2014.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. CRYPTO'10, pages 80–97, Berlin, Heidelberg, 2010. Springer-Verlag.

[PG13] Thomas Pöppelmann and Tim Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In *SAC*, 2013.

[RS10] Markus Rückert and Michael Schneider. Estimating the security of lattice-based cryptosystems. *IACR Cryptology ePrint Archive*, 2010:137, 2010.

[SS11] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47, 2011.

[SS13] Damien Stehlé and Ron Steinfeld. Making ntruencrypt and ntrusign as secure as standard worst-case problems over ideal lattices. *IACR Cryptology ePrint Archive*, 2013:4, 2013.

[Vau03] Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptology*, 16(4):249–286, 2003.