Lattice Signatures and Bimodal Gaussians

Léo Ducas[1], Alain Durmus[2,*], Tancrède Lepoint[3], and Vadim Lyubashevsky[4]

[1] ENS Paris, France
[2] ENPC and ENS Cachan, France
[3] CryptoExperts and ENS Paris, France
[4] INRIA and ENS Paris, France
{Leo.Ducas,Alain.Durmus,Tancrede.Lepoint,Vadim.Lyubashevsky}@ens.fr

Abstract. Our main result is a construction of a lattice-based digital signature scheme that represents an improvement, both in theory and in practice, over today's most efficient lattice schemes. The novel scheme is obtained as a result of a modification of the rejection sampling algorithm that is at the heart of Lyubashevsky's signature scheme (Eurocrypt, 2012) and several other lattice primitives. Our new rejection sampling algorithm which samples from a *bimodal* Gaussian distribution, combined with a modified scheme instantiation, ends up reducing the standard deviation of the resulting signatures by a factor that is asymptotically square root in the security parameter. The implementations of our signature scheme for security levels of 128, 160, and 192 bits compare very favorably to existing schemes such as RSA and ECDSA in terms of efficiency. In addition, the new scheme has shorter signature and public key sizes than all previously proposed lattice signature schemes.

As part of our implementation, we also designed several novel algorithms which could be of independent interest. Of particular note, is a new algorithm for efficiently generating discrete Gaussian samples over $\mathbb{Z}^n$. Current algorithms either require many high-precision floating point exponentiations or the storage of very large pre-computed tables, which makes them completely inappropriate for usage in constrained devices. Our sampling algorithm reduces the hard-coded table sizes from linear to logarithmic as compared to the time-optimal implementations, at the cost of being only a small factor slower.

1 Introduction

Lattice cryptography is arguably the most promising replacement for standard cryptography after the eventual coming of quantum computers. The most ubiquitous public-key cryptographic primitives, encryption schemes [18,26] and digital signatures [24,15], already have somewhat practical lattice-based instantiations. In addition, researchers are rapidly discovering new lattice-based primitives, such as fully-homomorphic encryption [10], multi-linear maps [9], and attribute-based encryption [14], that had no previous constructions based on classical number-theoretic techniques. Even though the above primitives are quite varied in their

* This work was done while the author was at ENS Paris, France.

R. Canetti and J.A. Garay (Eds.): CRYPTO 2013, Part I, LNCS 8042, pp. 40–56, 2013.

Table 1. Benchmarking on a desktop computer (Intel Core i7 at 3.4Ghz, 32GB RAM) with `openssl` 1.0.1c

	Security	Signature size	Sign (ms)	Sign/s	Verify (ms)	Verify/s
BLISS-0	$\leqslant$ 60 bits	3.3 kilobits	0.241	4k	0.017	59k
BLISS-I	128 bits	5.6 kb	0.124	8k	0.030	33k
BLISS-II	128 bits	5 kb	0.480	2k	0.030	33k
BLISS-III	160 bits	6 kb	0.203	5k	0.031	32k
BLISS-IV	192 bits	7 kb	0.375	2.5k	0.032	31k
RSA 1024	72-80 bits	1 kb	0.167	6k	0.004	91k
RSA 2048	103-112 bits	2 kb	1.180	0.8k	0.038	27k
RSA 4096	$\geqslant$ 128 bits	4 kb	8.660	0.1k	0.138	7.5k
ECDSA[1] **160**	80 bits	0.32 kb	0.058	17k	0.205	5k
ECDSA 256	128 bits	0.5 kb	0.106	9.5k	0.384	2.5k
ECDSA 384	192 bits	0.75 kb	0.195	5k	0.853	1k

functionalities, many of them share the same basic building blocks. Thus an improvement in one of these fundamental building blocks, usually results in the simultaneous improvement throughout lattice cryptography. For example, the recent work on the lattice trapdoor generation algorithm [27] resulted in immediate efficiency improvements in lattice-based hash-and-sign signatures, identity-based encryption schemes, group signatures, and functional encryption schemes.

In this work, we propose an improvement of another such building block – the *rejection sampling* procedure that is present in the most efficient constructions of lattice-based digital signatures [24,15], authentication schemes [23], blind signatures [31], and zero-knowledge proofs used in multi-party computation [4]. As a concrete application, we show that with our new algorithm, lattice-based digital signatures become completely practical. We construct and implement a family of digital signature schemes, named BLISS (Bimodal Lattice Signature Scheme) for security levels of 128, 160, and 192 bits. On standard 64-bit processors, our proof-of-concept implementations constitute significant improvements over previous lattice-based signatures and compare very favorably to the `openssl` implementations of RSA and ECDSA signatures schemes (see Table 1).

As part of our implementation, we also designed several novel algorithms that could be of independent interest. Chiefly among them is a new procedure that very efficiently samples from the Gaussian distribution over $\mathbb{Z}^m$ without requiring a very large look-up table. The absence of such an algorithm made researchers avoid using the Gaussian distribution when implementing lattice-based schemes on constrained devices, which resulted in these schemes being less compact than they could have been [15].

1.1 Related Work

Rejection Sampling. Rejection sampling in lattice constructions was first used by Lyubashevsky [22] to construct a three-round identification scheme. A

[1] ECDSA on a prime field $\mathbb{F}_p$: `ecdsap160`, `ecdsap256` and `ecdsap384` in `openssl`.

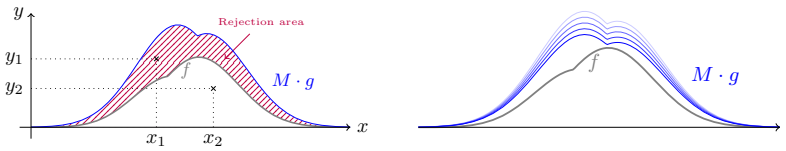
standard identification scheme is a three round sigma protocol that consists of a commit, challenge, and response stages. The main idea underlying their constructions and security proofs from number theoretic assumptions (e.g. Schnorr and GQ schemes [2]) is that the value y committed to in the first stage is used to information-theoretically hide the secret key s in the third stage. This is relatively straight-forward to do in number-theoretic schemes because one can just commit to a random y and then add it to (or multiply it by) some challenge-dependent function of s. Since all operations are performed in a finite ring, y being uniformly random hides s. In lattice constructions, however, we need to hide the secret key with a *small* y. The solution is thus to choose y from a narrow distribution and then perform rejection sampling so that s is not leaked when we add y to it (we describe this idea in much greater detail in Section 1.2). The improvements in lattice-based identification schemes (and therefore signature schemes via the Fiat-Shamir transformation) partly came via picking distributions that were more amenable to rejection sampling.

Lattice Signatures. Early lattice-based signature proposals did not have security reductions [13,19,17], and they were all subsequently broken because it turned out that every signature leaked a part of the secret key [12,29,6]. Among known provably-secure signature schemes, [11,23], [24,27], the most efficient seems to be that of [24] whose most efficient instantiation has both signature and key size of the order of 9kb [15] for approximately 80 bits of security.[2]

1.2 Our Results and Techniques

Rejection Sampling and Signature Construction. To understand our improvement of the rejection sampling procedure, we believe that it is useful to first give an overview of rejection sampling and the most efficient way in which it is currently used in constructing lattice-based signatures [24]. Rejection sampling is a well-known method introduced by von Neumann [33] to sample from an arbitrary target probability distribution f, given a source bound to a different probability distribution g. Conceptually, the method works as follows. A sample x is drawn from g and is accepted with probability $f(x)/(M \cdot g(x))$, where M is some positive real. If it is not accepted, then the process is restarted. It is not hard to prove that if $f(x) \leqslant M \cdot g(x)$ for all x, then the rejection sampling procedure produces exactly the distribution of f. Furthermore, because the expected number of times the procedure will need to be restarted is M, it is crucial to keep M as small as possible, possibly by tailoring the function g so that it resembles the target function f as much as possible. In particular, since rejection sampling can be interpreted as sampling a random point (x_i, y_i) in the area under the distribution $M \cdot g$ (see Figure 1) and accepting if and only if $y_i \leqslant f(x_i)$, reducing the area between the two curves will reduce M.

[2] In [15], a 100-bit security level was claimed, but the cryptanalysis we use in the full version of this paper [5], which combines lattice-reduction attacks with combinatorial meet-in-the-middle techniques [20], estimates the actual security to be around 75-80 bits.



(a) (x_i, y_i) is sampled uniformly in the area under $M \cdot g$, and accepted when $y_i \leqslant f(x_i)$

(b) M can be reduced when g is better adapted to f

Fig. 1. Rejection sampling from the distribution of g to get the distribution of f

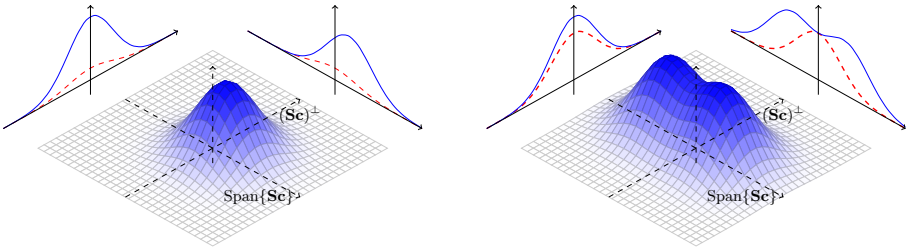
The digital signature from [24] works as follows (for the sake of this discussion, we will present the simplest version based on SIS): the secret key is an $m \times n$ matrix $\mathbf{S}$ with small coefficients, and the public key consists of a random $n \times m$ matrix $\mathbf{A}$ whose entries are uniform in $\mathbb{Z}_q$ and $\mathbf{T} = \mathbf{AS} \bmod q$. There is also a cryptographic hash function H, modeled as a random oracle, which outputs elements in $\mathbb{Z}^n$ with small norms. To sign a message digest μ, the signing algorithm first picks a vector $\mathbf{y}$ according to the distribution D_σ^m, where D_σ^m is the discrete Gaussian distribution over $\mathbb{Z}^m$ with standard deviation σ. The signer then computes $\mathbf{c} = H(\mathbf{Ay} \bmod q, \mu)$ and produces a potential signature $(\mathbf{z}, \mathbf{c})$ where $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$. Notice that the distribution of $\mathbf{z}$ depends on the distribution of $\mathbf{Sc}$, and thus on the distribution of $\mathbf{S}$ – in fact, the distribution of $\mathbf{z}$ is exactly D_σ^m *shifted* by the vector $\mathbf{Sc}$.

To remove the dependence of the signature on $\mathbf{S}$, rejection sampling is used. The target distribution that we want for signatures is D_σ^m, whereas we obtain samples from the distribution D_σ^m shifted by $\mathbf{Sc}$ (call this distribution $D_{\mathbf{Sc},\sigma}^m$). To use rejection sampling, we need to find a positive real M such that for all (or all but a negligible fraction) $\mathbf{x}$ distributed according to D_σ^m we have $D_\sigma^m(\mathbf{x}) \leqslant M \cdot D_{\mathbf{Sc},\sigma}^m(\mathbf{x})$. A simple calculation (see [24, Lemma 4.5]) shows that

$$D_\sigma^m(\mathbf{x}) / D_{\mathbf{Sc},\sigma}^m(\mathbf{x}) = \exp\left(\frac{-2\langle \mathbf{x}, \mathbf{Sc}\rangle + \|\mathbf{Sc}\|^2}{2\sigma^2}\right). \tag{1}$$

The value of $\langle \mathbf{x}, \mathbf{Sc}\rangle$ behaves in many ways as a one-dimensional discrete Gaussian, and it can be thus shown that $|\langle \mathbf{x}, \mathbf{Sc}\rangle| < \tau\sigma\|\mathbf{Sc}\|$ with probability $1 - \exp(-\Omega(\tau^2))$. Asymptotically, the value of τ is proportional to the square root of the security parameter. Concretely, if we would like to have, for example, $1-2^{-100}$ certainty that $|\langle \mathbf{x}, \mathbf{Sc}\rangle| < \tau\sigma\|\mathbf{Sc}\|$, we would set $\tau = 12$. Thus with probability $1 - \exp(-\Omega(\tau^2))$, we have $\exp\left(\frac{-2\langle \mathbf{x},\mathbf{Sc}\rangle + \|\mathbf{Sc}\|^2}{2\sigma^2}\right) \leqslant \exp\left(\frac{2\tau\sigma\|\mathbf{Sc}\| + \|\mathbf{Sc}\|^2}{2\sigma^2}\right)$. So if $\sigma = \tau\|\mathbf{Sc}\|$, we will have $D_\sigma^m(\mathbf{x}) / D_{\mathbf{Sc},\sigma}^m(\mathbf{x}) \leqslant \exp\left(1 + \frac{1}{2\tau^2}\right)$. Therefore if we set $M = \exp\left(1 + \frac{1}{2\tau^2}\right)$, rejection sampling outputs signatures that are distributed according to D_σ^m where $\sigma = \tau\|\mathbf{Sc}\|$ and the expected number of repetitions is $M \approx \exp(1)$.[3]

[3] More precisely $\sigma = \tau \max_{\mathbf{S},\mathbf{c}} \|\mathbf{Sc}\|$, since $\mathbf{Sc}$ is not known in advance.



(a) In the original scheme of [24] (b) In our scheme

Fig. 2. Improvement of Rejection Sampling with Bimodal Gaussian Distributions. In blue is the distribution of $\mathbf{z}$, for fixed $\mathbf{Sc}$ and over the space of all $\mathbf{y}$ in Figure (a) and all $(b, \mathbf{y})$ in Figure (b), before the rejection step and its decomposition as a Cartesian product over $\mathrm{Span}\{\mathbf{Sc}\}$ and $(\mathbf{Sc})^{\perp}$. In dashed red is the target distribution scaled by $1/M$.

Prior to explaining our technique to improve the scheme, we need to state how the verification algorithm in [24] works. Upon receiving the signature $(\mathbf{z}, \mathbf{c})$ of μ, the verifier checks that $\|\mathbf{z}\|$ is "small" (roughly $\sigma\sqrt{m}$) and also that $\mathbf{c} = H(\mathbf{Az} - \mathbf{Tc} \bmod q, \mu)$. It is easy to check that the outputs of the signing procedure satisfy the two requirements. In this work, we show how to remove the factor τ (in fact even more) from the required standard deviation. Above, we described how to perform rejection sampling when we were sampling potential signatures as $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$. Consider now, an alternative procedure, where we first uniformly sample a bit $b \in \{-1, 1\}$ and then choose the potential signature to be $\mathbf{z} = b\mathbf{Sc} + \mathbf{y}$. In particular $\mathbf{z}$ is now sampled from the distribution $\frac{1}{2}D^m_{\mathbf{Sc},\sigma} + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}$. If our target distribution is still D^m_σ, then, as above, we need to have $D^m_\sigma(\mathbf{x}) / \left(\frac{1}{2}D^m_{\mathbf{Sc},\sigma}(\mathbf{x}) + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}(\mathbf{x})\right) \leqslant M$. By using Equation (1) and some algebraic manipulations, we obtain that

$$D^m_\sigma(\mathbf{x}) / \left(\frac{1}{2}D^m_{\mathbf{Sc},\sigma}(\mathbf{x}) + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}(\mathbf{x})\right) = \exp\left(\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) / \cosh\left(\frac{\langle \mathbf{x}, \mathbf{Sc}\rangle}{\sigma^2}\right) \leqslant \exp\left(\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right),$$

where the last inequality follows from the fact that $\cosh(y) \geqslant 1$ for all y. Thus for rejection sampling to work with $M = \exp(1)$, as in the previous example, we only require that $\sigma = \|\mathbf{Sc}\|/\sqrt{2}$ rather than $\tau\|\mathbf{Sc}\|$.

Our improvement is depicted on Figure 2. Part 2(a) shows the rejection sampling as done in [24]. There, the distribution D^m_σ (the dashed red line) must be scaled by a somewhat large factor so that all but a negligible fraction of it fits under $D^m_{\mathbf{Sc},\sigma}$. In 2(b), which represents our improved sampling algorithm, the distribution from which we are sampling is bimodal having its two centers at $\mathbf{Sc}$ and $-\mathbf{Sc}$. As can be seen from the figure, the distribution D^m_σ fits much "better" (*i.e.* needs

to be scaled by a much smaller factor) underneath the bimodal distribution and therefore there is a much smaller rejection area between the two curves. As a side note, whereas in (a), a negligible fraction of the scaled D_σ^m is still above $D_{\mathbf{Sc},\sigma}^m$, in (b), all of D_σ^m is underneath the bimodal distribution $\frac{1}{2}D_{\mathbf{Sc},\sigma}^m + \frac{1}{2}D_{-\mathbf{Sc},\sigma}^m$.

While the above sampling procedure potentially produces much shorter signatures since the Gaussian "tail-cut" factor τ is never used, it does not give an improved signature scheme by itself because the verification procedure is no longer guaranteed to work. The verification checks that $\mathbf{c} = H(\mathbf{Az}-\mathbf{Tc} \bmod q, \mu)$ and so will verify correctly if and only if $\mathbf{Ay} = \mathbf{Az} - \mathbf{Tc} = \mathbf{A}(b\mathbf{Sc} + \mathbf{y}) - \mathbf{Tc} = \mathbf{Ay} + b\mathbf{Tc} - \mathbf{Tc}$, which will only happen if $b\mathbf{Tc} = \mathbf{Tc} \bmod q$ for $b \in \{-1, 1\}$. In other words, we will need $\mathbf{Tc} = -\mathbf{Tc} \bmod q$, which will never happen if q is prime unless $\mathbf{T} = \mathbf{0}$. [4] Our solution, therefore, is to work modulo $2q$ and to set $\mathbf{T} = q\mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix. In this case $\mathbf{Tc} = -\mathbf{Tc} \bmod 2q$, and so the verification procedure will always work.

Changing the modulus from q to $2q$ and forcing the matrix $\mathbf{T}$ to always be $q\mathbf{I}$ creates several potential problems. In particular, it is no longer clear how to perform key generation, and also the outline for the security proof from [24] no longer holds. But we show that these problems can be overcome. We will now sketch the key generation and the security proof based on the hardness of the SIS problem in which one is given a uniformly random matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, and is asked to find a short vector $\mathbf{w}$ such that $\mathbf{Bw} = \mathbf{0} \pmod q$. To generate the public and secret keys, we first pick a uniformly random matrix $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}$ and a random $(m-n) \times n$ matrix $\mathbf{S}'$ consisting of short coefficients. We then compute $\mathbf{A}'' = \mathbf{A}'\mathbf{S}' \bmod q$ and output $\mathbf{A} = [2\mathbf{A}'|2\mathbf{A}'' + q\mathbf{I}]$ as the public key. The secret key is $\mathbf{S} = [\mathbf{S}'| - \mathbf{I}]^T$. Notice that by construction we have $\mathbf{AS} = q\mathbf{I} \pmod{2q}$ and $\mathbf{S}$ consists of small entries. The dimensions m and n are picked so that the distribution of $[\mathbf{A}'|\mathbf{A}'\mathbf{S}' \bmod q]$ can be shown to be uniformly random in $\mathbb{Z}_q^{n \times m}$ by the leftover hash lemma.

In the security proof, we are given a random matrix $\mathbf{B} = [\mathbf{A}'|\mathbf{A}''] \in \mathbb{Z}_q^{n \times m}$ by the challenger and use the adversary that forges a signature to find a short vector $\mathbf{w}$ such that $\mathbf{Bw} = \mathbf{0} \pmod q$. We create the public key $\mathbf{A} = [2\mathbf{A}'|2\mathbf{A}'' + q\mathbf{I}]$ and give it to the adversary. Even though we do not know a secret key $\mathbf{S}$ such that $\mathbf{AS} = q\mathbf{I} \pmod{2q}$, we can still create valid signatures for any messages of the adversary's choosing by picking the $(\mathbf{z}, \mathbf{c})$ according to the correct distributions and then programming the random oracle as is done in [24]. When the adversary forges, we use the forking lemma to create two equations $\mathbf{Az} = q\mathbf{c} \pmod{2q}$ and $\mathbf{Az}' = q\mathbf{c}' \pmod{2q}$. Combining them together, we obtain $\mathbf{A}(\mathbf{z} - \mathbf{z}') = q(\mathbf{c} - \mathbf{c}') \pmod{2q}$. Under some very simple requirements for $\mathbf{z}, \mathbf{z}', \mathbf{c}$, and $\mathbf{c}'$, the previous equation implies that $\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{0} \pmod q$ and $\mathbf{z} \neq \mathbf{z}'$. This then implies that $2\mathbf{B}(\mathbf{z} - \mathbf{z}') = \mathbf{0} \pmod q$ and since 2 is invertible modulo q, we have found a $\mathbf{w} = (\mathbf{z} - \mathbf{z}')$ such that $\mathbf{Bw} = \mathbf{0} \pmod q$.

[4] One may think that a possible solution could be to output the bit b as part of the signature, but this is not secure. Depending on the sign of $\langle \mathbf{z}, \mathbf{Sc} \rangle$, one of the two values of b is more likely to be output than the other. Therefore outputting the bit b leaks information about $\mathbf{S}$.

The above scheme construction and proof work for SIS and equally well for Ring-SIS, when instantiated with polynomials. As in [24], we can also construct much more efficient schemes based on LWE and Ring-LWE by creating the matrix $\mathbf{A}'' = \mathbf{A}'\mathbf{S}'$ such that $(\mathbf{A}', \mathbf{A}'')$ is not uniformly random, but only computationally. For optimal efficiency, though, we can create the key in yet a different manner related to the way NTRU keys are generated. The formal construction is described in the full version, and we just give the intuition here. We could create two small polynomials $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}[x]/(x^n + 1)$ and output the public key as $\mathbf{a} = \frac{\mathbf{q}-\mathbf{s}_2}{\mathbf{s}_1} \pmod{2q}$. Notice that this implies that $\mathbf{a}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{q} \pmod{2q}$, and so we can think of the public key as $\mathbf{A} = [\mathbf{a}, \mathbf{1}]$ and the secret key as $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2]^T$. Assuming that it is a hard problem to find small vectors $\mathbf{w}$ such that $\mathbf{A}\mathbf{w} = \mathbf{0} \pmod{2q}$, the signature scheme instantiated in the above manner will be secure. To those readers familiar with the key generation in the NTRU encryption scheme, the above key generation should look very familiar, except that the modulus is $2q$ rather than q. Since we are not sure what happens when the modulus is $2q$, we show in the full version of this paper [5] how to instantiate our scheme so that it is based on NTRU over modulus q. We then explain how for certain instantiations, this is as hard a problem as Ring-SIS (using the results of Stehlé, Steinfeld [32]) and how for more efficient instantiations, it is a *weaker* assumption than the ones underlying the classic NTRU encryption scheme and the recent construction of fully-homomorphic encryption [21].

Gaussian Sampling. There are two generic methods for sampling according to a discrete Gaussian distribution. The first one uses basic rejection sampling as follows: choose a uniform integer $x \in \{-\tau\sigma, \ldots, \tau\sigma\}$ (where $\tau \approx 12$, as in the preceding discussion) and accept it with probability proportional to $\exp(-x^2/2\sigma^2)$ (and restart otherwise). This involves the computation of the exp function to high precision and requires an average of $2\tau/\sqrt{2\pi} \approx 10$ trials, thus wasting a lot of random bits. The second one involves storing large pre-computed data, namely the cumulative distribution table of the discrete Gaussian from $-\tau\sigma$ to $\tau\sigma$. While the second method is very efficient when given enough memory, neither of the two approaches is appropriate for use in constrained devices.

We solve this issue by modifying the first approach to exploit the properties of discrete Gaussians. We recall that a Bernoulli distribution $\mathcal{B}_c$ assigns 1 (True) with probability $c \in [0, 1]$ and 0 (False) with probability $1 - c$. Overloading the notation for the sake of clarity, we will denote by $\mathcal{B}_c$ both the distribution and a generic random variable that follows that distribution independently of all others (thus we may write, for example, $\mathcal{B}_a \oplus \mathcal{B}_b = \mathcal{B}_{a+b-2ab}$). As a first step, to avoid explicit computation of exp, we use the simple fact that for an integer x in binary form $x = x_1 \cdots x_n$ we have $\mathcal{B}_{\exp(-x/f)} = \bigwedge_{i \text{ s.t. } x_i=1} \mathcal{B}_{\exp(-2^i/f)}$. This allows us to sample according to $\mathcal{B}_{\exp(-x/f)}$ using only logarithmically many precomputed values $\exp(-2^i/f)$. Similarly, we also design another algorithm to sample according to $\mathcal{B}_{1/\cosh(x/f)}$, using a Markov chain that makes less than two calls to $\mathcal{B}_{\exp(-x/f)}$ on average.

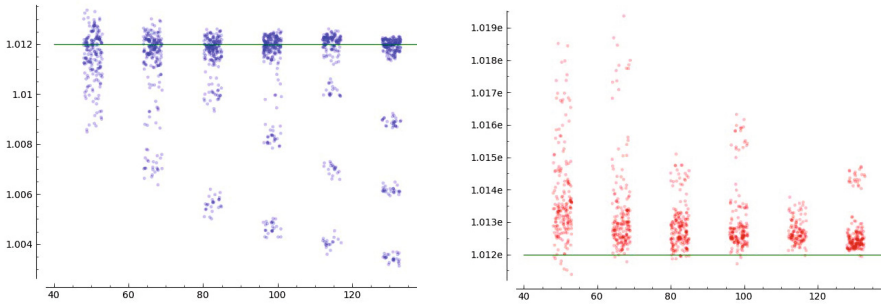
The second step is to replace the uniform distribution from which one chooses an integer by a more adapted one to decrease the rejection rate. It is essential, though, that the rejection rate retains an easily samplable form. To do this, we build on a specific discrete Gaussian of variance $\sigma_2^2 = 1/(2\ln 2)$ for which the distribution $D_{\sigma_2}(x)$ is proportional to 2^{-x^2}. This makes it very easily samplable, and the rejection rate still has the required form $\exp(\cdot/f)$. The final algorithm has bounded repetition rate of 1.5 rather than $2\tau/\sqrt{2\pi} \approx 10$. All the operations are very simple, requiring only small integer arithmetic, and are therefore well-suited for constrained devices.

Cryptanalysis and Experiments on NTRU Lattices. Previous cryptanalytic efforts against schemes based on SIS and LWE mostly involved computing the Hermite factor of the underlying average-case instance, as in the work of Gama and Nguyen [8], and making sure that its value is below the level required for the desired security guarantees. In this work (described in detail in the full version of this paper [5]) we undertake a more careful cryptanalysis by using the results on BKZ 2.0 of Chen and Nguyen [3] in combination with other techniques – namely dual lattice reduction and the combinatorial meet-in-the-middle attack of Howgrave-Graham [20].

For optimal efficiency, the security of our scheme relies on the hardness of a type of NTRU problem that has recently (re-)appeared in the literature [21] and which, we believe, could play a major role in the future of lattice-based cryptography (see Section 2 for the precise definition of the problem). The only cryptanalysis of which we are aware of that studies NTRU lattices deals with instances where the modulus is very close in size to the dimension of the lattice [8,16]. It is thus unclear as to what roles each of the variables plays when looked at independently.

In our work, and also in the previously-mentioned work of [21], the modulus is required to be substantially larger than the dimension. As far as we are aware, no previous cryptanalysis was done for these types of instances. The most complete study of the behavior of BKZ in the presence of unusually short vector(s) is due to Gama and Nguyen [8] who thoroughly analyzed the algorithm's running time in the presence of *one* such vector. Their experiments show that the hardness of finding this vector depends on the ratio λ_2/λ_1, that is, the gap between the second-shortest and the shortest vectors in the m-dimensional lattice. In practice, for BKZ-20, the shortest vector was found when $\lambda_2/\lambda_1 > .48 \cdot 1.01^m$.

We ran similar experiment of BKZ-20 in the case of $2n$-dimensional NTRU lattices where $\lambda_1 = \ldots = \lambda_n$. In NTRU lattices, the gap normally occurs between the n-th and the $n+1$-st successive minima, and one might think that the ratio between these two quantities would somehow determine the hardness of the instance. Our experiments showed that this is not the case, and the shortest vector was found when $\sqrt{qm/2\pi e}/\lambda_1$ was greater than $.40 \cdot 1.012^m$ (see Figure 3). Despite the fact that there is no vector in the lattice having length $\sqrt{qm/2\pi e}$ this is actually consistent with the results of [8]! The reason is that $\sqrt{qm/2\pi e}$ is



(a) Shortest vector not found (b) Shortest vector found

Fig. 3. Results BKZ-20 for $n \in [48, 150]$, $q \in [6000, 25000]$ and binary search on the λ_1-threshold. On horizontal axis is the value of $n + \texttt{random(0,5)}$ and on vertical axis is $\left(\frac{1}{.40}\sqrt{\frac{qm}{2\pi e}}/\lambda_1\right)^{1/2n}$.

the *expected* length of the shortest vector according to the Gaussian heuristic,[5] and we would also expect $\lambda_2 \approx \sqrt{qm/2\pi e}$ in a random q-ary lattice analyzed in [8]. Thus one could say that the hardness of finding a short vector in q-ary lattices depends not on the gap, but rather on the ratio between the Gaussian heuristic and the actual length of the shortest vector.

Similar to the results in [8], when the ratio was smaller than $.40 \cdot 1.012^m$, the resulting shortest vector had length about $\sqrt{q} \cdot 1.012^m$. In other words, BKZ-20 behaved as if the lattice were truly random. Because of our experiments with BKZ-20, it seems reasonable to assume that BKZ behaves analogously for larger block sizes. Thus we can measure its efficacy according to the BKZ 2.0 methodology in [3]. We would like to stress that we have no explanation for the reason why the ratio between the Gaussian heuristic and the actual length of the vector seems to dictate the hardness of finding short vectors in NTRU lattices. We are equally unsure whether this phenomenon implies that these lattices are indeed as hard as the random lattices that have been more exhaustively studied [8,3].

The general dearth of lattice cryptanalysis papers stands in contrast to the vast number of articles proposing theoretical lattice-based constructions. Our belief is that this lack of cryptanalytic effort is in part due to the fact that most of the papers with scheme proposals give no concrete targets to attack. One of the proposed instantiations in the present work is a "toy example" that we estimate has approximately 60 bits of security. Thus if it turns out that NTRU lattices are weaker than believed, it is wholly possible that this example could be broken on a personal computer, and we think this would be of great interest to the practical community. In addition, it could be argued that we do not yet know enough about lattice reduction to be able to propose such "fine-grained" security estimates like 160-bit or 192-bit. But one of the main reasons

[5] The Gaussian heuristic says that for certain types of random lattices $\mathcal{L}$, we will have $\lambda_1(\mathcal{L}) \approx det(\mathcal{L})^{1/m} \cdot \sqrt{\frac{m}{2\pi e}}$ [8].

that we make these proposals is to make it "worthwhile" for cryptanalysts to work on these problems. In short, one of our hopes is that this work spurs on the cryptanalysis that is currently much needed in the field.

Acknowledgments. We thank the CRYPTO 2013 reviewers for their careful reading of the paper and their diligent comments. We also thank Steven Galbraith and Pascal Paillier for useful comments on previous versions of this work.

2 Preliminaries

2.1 Notation

For any integer q, we identify the ring $\mathbb{Z}_q$ with the interval $[-q/2, q/2) \cap \mathbb{Z}$, and in general for a ring $\mathcal{R}$, we define $\mathcal{R}_q$ to be the quotient ring $\mathcal{R}/(q\mathcal{R})$. Whenever working in the quotient ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$, we will assume that n is a power of 2 and q is a prime number such that $q = 1 \pmod{2n}$. Vectors, considered as column vectors, will be written in bold lower case letters. Matrices will be written in bold upper case letters. For a positive integer n, we write $\mathbf{I}_n$ to be the identity matrix of dimension n.

We recall that the ℓ_p-norm of a vector $\mathbf{v}$ is defined as $\|\mathbf{v}\|_p = (\sum_i |v_i|^p)^{1/p}$ for $p > 0$, and its ℓ_∞-norm as $\|\mathbf{v}\|_\infty = \max_i |v_i|$. By default, we use $\|\cdot\|$ for the ℓ_2-norm.

We now state a general rejection sampling lemma. The proof of this lemma is quite standard (cf. [24]).

Lemma 2.1 (Rejection Sampling). *Let V be an arbitrary set, and $h\colon V \to \mathbb{R}$ and $f\colon \mathbb{Z}^m \to R$ be probability distributions. If $g_v\colon \mathbb{Z}^m \to R$ is a family of probability distributions indexed by $v \in V$ with the property that there exists a $M \in \mathbb{R}$ such that*

$$\forall v \in V, \forall \mathbf{z} \in \mathbb{Z}^m, M \cdot g_v(\mathbf{z}) \geqslant f(\mathbf{z}) \ ,$$

then, the output distributions of the following two algorithms are identical:

1. *$v \leftarrow h$, $z \leftarrow g_v$, output $(\mathbf{z}, v)$ with probability $f(\mathbf{z})/\big(M \cdot g_v(\mathbf{z})\big)$.*
2. *$v \leftarrow h$, $z \leftarrow f$, output $(\mathbf{z}, v)$ with probability $1/M$.*

2.2 Discrete Gaussian Distribution

Gaussian Distribution. The (un-normalized) Gaussian distribution with standard deviation $\sigma \in \mathbb{R}$ and center $c \in \mathbb{R}$ evaluated at $x \in \mathbb{R}$ is defined by $\rho_{c,\sigma}(x) = \exp\big(\frac{-(x-c)^2}{2\sigma^2}\big)$, and more generally by $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp\big(\frac{-\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}\big)$ for $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$. When the center $\mathbf{c}$ is $\mathbf{0}$, we generally omit it from the notation and simply write $\rho_\sigma(\mathbf{x})$. The discrete Gaussian distribution over $\mathbb{Z}$ centered at 0 is defined by $D_\sigma(x) = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$, and more generally, over $\mathbb{Z}^m$ by $D^m_\sigma(\mathbf{x}) = \rho_\sigma(\mathbf{x})/\rho_\sigma(\mathbb{Z})^m$.

Tailcutting. It is generally useful to ignore large values which are unlikely to appear when drawing according to a Gaussian distribution.

Lemma 2.2 ([28]). *For any dimension m, $\sigma > 0$ and $\tau > 1$, $\rho_\sigma(\mathbb{Z}^m \setminus \tau\sigma\sqrt{m}\mathfrak{B}) < 2C(\tau)^m \cdot \rho_\sigma(\mathbb{Z})^m$, where $C(\tau) = \tau \exp\left(\frac{1-\tau^2}{2}\right) < 1$, and $\mathfrak{B}$ is the centered ℓ_2 unit ball.*

Therefore, to tailcut less than $2^{-\lambda}$ of a 1-dimensional Gaussian, one should choose $\tau \approx \sqrt{\lambda \cdot 2 \ln 2}$, the typical value being $\tau = 12$ for $\lambda = 100$.

2.3 Hardness Assumptions

All the constructions in this paper are based on the hardness of the *generalized SIS* (Short Integer Solution) problem, which we define below.

Definition 2.3 ($\mathcal{R}$-SIS$^{\mathcal{K}}_{q,n,m,\beta}$ problem). *Let $\mathcal{R}$ be some ring and $\mathcal{K}$ be some distribution over $\mathcal{R}_q^{n\times m}$, where $\mathcal{R}_q$ is the quotient ring $\mathcal{R}/(q\mathcal{R})$. Given a random $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ drawn according to the distribution $\mathcal{K}$, find a non-zero $\mathbf{v} \in \mathcal{R}_q^m$ such that $\mathbf{A}\mathbf{v} = \mathbf{0}$ and $\|\mathbf{v}\|_2 \leqslant \beta$.*

If we let $\mathcal{R} = \mathbb{Z}$ and $\mathcal{K}$ be the uniform distribution, then the resulting problem is the classical SIS problem first defined by Ajtai [1] in his seminal paper showing connections between worst-case lattice problems and the average-case SIS problem. By the pigeonhole principle, if $\beta \geqslant \sqrt{m}q^{n/m}$ then the SIS instances are guaranteed to have a solution. Using Gaussian techniques, Micciancio and Regev [28] improved Ajtai's result to show that, for a large enough q as a function of n and β, the SIS$_{q,n,m,\beta}$ problem is as hard (on the average) as the $\tilde{\mathcal{O}}(\sqrt{n}\beta)$-SIVP problem for *all* lattices of dimension n.

In 2006, a ring variant of SIS was introduced independently by Peikert and Rosen [30] and Lyubashevsky and Micciancio [25]. In [25] it was shown that if $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$, where n is a power of 2, then the $\mathcal{R}$-SIS$^{\mathcal{K}}_{q,1,m,\beta}$ problem is as hard as the $\tilde{\mathcal{O}}(\sqrt{n}\beta)$-SVP problem in all lattices that are ideals in $\mathcal{R}$ (where $\mathcal{K}$ is again the uniform distribution over $\mathcal{R}_q^{1\times m}$).

NTRU Lattices. In the NTRU cryptosystem over the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ [18], the key generation procedure picks two short secret keys $\mathbf{f}, \mathbf{g} \in \mathcal{R}_q$ (according to some distribution) and computes the public key as $\mathbf{a} = \mathbf{g}/\mathbf{f}$.[6] When the norm of $\mathbf{f}, \mathbf{g}$ is large enough, it can be shown that $\mathbf{a}$ is actually uniformly random in $\mathcal{R}_q$ [32], but even when the secret keys do not have enough entropy, their quotient still appears to be pseudorandom, although no proof of this fact is known [21]. In the NTRU cryptosystem (or its more secure modification of [32] which is based on the Ring-LWE problem), one encrypts a message μ, represented as a polynomial in $\mathcal{R}_q$ with $\{0, 1\}$ coefficients, by picking two short vectors

[6] In the original NTRU scheme, the ring was $\mathbb{Z}_q[x]/(x^n - 1)$, but lately researchers have also used $\mathbb{Z}_q[x]/(x^n + 1)$ when n is a power of 2. Indeed, the latter choice seems at least as secure.

$\mathbf{r}, \mathbf{e} \in \mathcal{R}_q$ and outputting $\mathbf{z} = 2(\mathbf{ar} + \mathbf{e}) + \mu$. The security of the scheme relies on the fact that the distribution of $(\mathbf{a}, \mathbf{z})$ is pseudo-random in $\mathcal{R}_q^2$.

One can define an NTRU version of the SIS problem that is at least as hard as breaking the NTRU cryptosystem.[7] In particular, given an NTRU public key $\mathbf{a}$, find two polynomials $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{R}_q$ such that $\|(\mathbf{v}_1|\mathbf{v}_2)\| \leqslant \beta$ and $\mathbf{av}_1 + \mathbf{v}_2 = 0$ in $\mathcal{R}_q$. Notice that $(\mathbf{f}, -\mathbf{g})$ is a solution to this problem, but in fact, finding larger solutions can also be useful in breaking the NTRU cryptosystem. In particular, notice that for any solution $(\mathbf{v}_1|\mathbf{v}_2)$, one can compute $\mathbf{zv}_1 = 2(-\mathbf{rv}_2 + \mathbf{ev}_1) + \mu\mathbf{v}_1$. If β is sufficiently small with respect to $\|(\mathbf{r}|\mathbf{e})\|$, then $\mathbf{z} \cdot \mathbf{v}_1 \bmod 2 = \mu\mathbf{v}_1$, and μ can be recovered. Thus, for certain parameters, the NTRU version of the SIS problem is at least as hard as breaking the NTRU cryptosystem. As a side-note, we would like to point out that the NTRU encryption scheme remains hard even after 15 years of cryptanalysis. The weakness in the NTRU signature scheme, which uses the same key generation procedure, is due to the fact that signatures slowly leak the secret key [29,6], but this is provably (*i.e.* information-theoretically) avoided in our scheme.

In the full version of this paper [5], we propose a practical instantiation of our signature scheme inspired by the NTRU key-generation, and analyze the hardness of the NTRU version of the SIS problem using combinations of lattice [3] and hybrid attacks [20]. We provide concrete parameters, and the resulting signature scheme was implemented as a proof-of-concept on a desktop computer (and yielded the timings of Table 1).

3 BLISS: A Lattice Signature Scheme Using Bimodal Gaussians

In this section, we present our new signature scheme along with the proof of correctness. The security of the signature scheme is based on the hardness of the $\mathcal{R}\text{-SIS}^{\mathcal{K}}_{q,n,m,\beta}$ problem. We mention that this is the "simple" version of our algorithm, and its more optimized implementation that uses numerous enhancements is presented in the full version of this paper [5]. For simplicity, we present our algorithm for $\mathcal{R} = \mathbb{Z}$, but it works in exactly the same way for rings $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$.

3.1 New Signature and Verification Algorithms

Key pairs. The secret key is a (short) matrix $\mathbf{S} \in \mathbb{Z}_{2q}^{m \times n}$ and the public key is given by the matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n \times m}$ such that $\mathbf{AS} = q\mathbf{I}_n \pmod{2q}$. A crucial property, for our new rejection sampling algorithm, satisfied by the key pair, is that $\mathbf{AS} = \mathbf{A}(-\mathbf{S}) = q\mathbf{I}_n \pmod{2q}$. Obtaining such a key pair is easy and can be done efficiently. In the full version of this paper [5], we explain the key-generation procedure which results in a scheme whose security is based on the

[7] A way to state the NTRU SIS problem in terms of the $\mathcal{R}\text{-SIS}^{\mathcal{K}}_{q,1,2,\beta}$ problem is to set $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ and let $\mathcal{K}$ be the distribution that picks small $\mathbf{f}, \mathbf{g}$ and outputs the public key $\mathbf{A} = (\mathbf{a}, \mathbf{1}) \in \mathcal{R}_q^{1 \times 2}$ for $\mathbf{a} = \mathbf{g}/\mathbf{f}$.

classic $\mathrm{SIS}_{q,n,m,\beta}$ problem and we present an "NTRU-like" variant of the key generation which yields a more efficient instantiation of the signature scheme.

Random Oracle Domain. We model the hash function H as a random oracle that has uniform output in $\mathbb{B}_\kappa^n$, the set of binary vectors of length n and weight κ. Such a mapping can be found in [7] but its complexity is quadratic in n; in the full version of this paper, we provide an efficient construction.

Algorithm 1. Signature Algorithm

Input: Message μ, public key $\mathbf{A} \in \mathbb{Z}_{2q}^{n \times m}$, secret key $\mathbf{S} \in \mathbb{Z}_{2q}^{m \times n}$, stand. dev. $\sigma \in \mathbb{R}$
Output: A signature $(\mathbf{z}, \mathbf{c})$ of the message μ

1: $\mathbf{y} \leftarrow D_\sigma^m$
2: $\mathbf{c} \leftarrow H(\mathbf{A}\mathbf{y} \bmod 2q, \mu)$
3: Choose a random bit $b \in \{0, 1\}$
4: $\mathbf{z} \leftarrow \mathbf{y} + (-1)^b \mathbf{S}\mathbf{c}$
5: **Output**$(\mathbf{z}, \mathbf{c})$ with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{S}\mathbf{c}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle}{\sigma^2}\right)\right)$ otherwise **restart**

Algorithm 2. Verification Algorithm

Input: Message μ, public Key $\mathbf{A} \in \mathbb{Z}_{2q}^n$, signature $(\mathbf{z}, \mathbf{c})$
Output: Accept or Reject the signature

1: **if** $\|\mathbf{z}\| \quad > B_2$ **then** Reject
2: **if** $\|\mathbf{z}\|_\infty \geqslant q/4$ **then** Reject
3: Accept iff $\mathbf{c} = H(\mathbf{A}\mathbf{z} + q\mathbf{c} \bmod 2q, \mu)$

The Signature Algorithm. The signer, who is given a message digest μ, first samples a vector $\mathbf{y}$ from the m-dimensional discrete Gaussian distribution D_σ^m and then computes $\mathbf{c} \leftarrow H(\mathbf{A}\mathbf{y} \bmod 2q, \mu)$. He then samples a bit b in $\{0, 1\}$ and computes the potential output $\mathbf{z} \leftarrow \mathbf{y} + (-1)^b \mathbf{S}\mathbf{c}$. Notice that $\mathbf{z}$ is distributed according to the bimodal discrete Gaussian distribution $\frac{1}{2} D_{\mathbf{S}\mathbf{c},\sigma}^m + \frac{1}{2} D_{-\mathbf{S}\mathbf{c},\sigma}^m$. At this point we perform rejection sampling and output the signature $(\mathbf{z}, \mathbf{c})$ with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{S}\mathbf{c}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle}{\sigma^2}\right)\right)$, where M is some fixed positive real that is set large enough to ensure that the preceding probability is always at most 1. We explain how to set M in accordance with the standard deviation σ in the next section. If the signing algorithm did not output the signature, then it is restarted and repeated until something is output. The expected number of iterations of the signing algorithm is M.

The Verification Algorithm. The verification algorithm will accept $(\mathbf{z}, \mathbf{c})$ as the signature for μ if the following three conditions hold:

1. $\|\mathbf{z}\| \quad \leqslant B_2$
2. $\|\mathbf{z}\|_\infty < q/4$
3. $\mathbf{c} = H(\mathbf{A}\mathbf{z} + q\mathbf{c} \bmod 2q, \mu)$

The signer outputs signatures of the form $(\mathbf{z}, \mathbf{c})$ where $\mathbf{z}$ is distributed according to D_σ^m, thus the acceptance bound B_2 should be set a little bit higher than $\sqrt{m}\sigma$, which is the expected value around which the output of D_σ^m is tightly concentrated; denoting $B_2 = \eta\sqrt{m}\sigma$, one can set η so that $\|\mathbf{z}\| \leqslant B_2$ is verified with probability $1-2^{-\lambda}$ [24, Lemma 4.4] for the security parameter λ (in practice, $\eta \in [1.1, 1.4]$). For technical reasons in the security proof, we also need that $\|\mathbf{z}\|_\infty < q/4$, but this condition is usually verified whenever the first one is and does not restrict the manner in which we choose the parameters for the scheme. Condition 3 will also hold for valid signatures because

$$\begin{aligned}\mathbf{Az} + q\mathbf{c} &= \mathbf{A}(\mathbf{y} + (-1)^b\mathbf{Sc}) + q\mathbf{c} = \mathbf{Ay} + \big((-1)^b\mathbf{AS}\big)\mathbf{c} + q\mathbf{c} = \mathbf{Ay} + (q\mathbf{I}_n)\mathbf{c} + q\mathbf{c} \\ &= \mathbf{Ay} \bmod 2q.\end{aligned}$$

3.2 Rejection Sampling: Correctness and Efficiency

We now explain how to pick the standard deviation σ and positive real M so that the signing algorithm in the preceding section produces vectors $\mathbf{z}$ according to the distribution D_σ^m. Because $\mathbf{y}$ is distributed according to D_σ^m, it is easy to see that in Step 4 of the signing algorithm, $\mathbf{z}$ is distributed according to $g_{\mathbf{Sc}} = \frac{1}{2}D^m_{\mathbf{Sc},\sigma} + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}$ for fixed $\mathbf{Sc}$ and over the space of all $(b, \mathbf{y})$. Thus for any $\mathbf{z}^* \in \mathbb{R}^m$, we have

$$\begin{aligned}\Pr[\mathbf{z} = \mathbf{z}^*] &= \frac{1}{2}D^m_{\mathbf{Sc},\sigma}(\mathbf{z}^*) + \frac{1}{2}D^m_{-\mathbf{Sc},\sigma}(\mathbf{z}^*) \\ &= \frac{1}{2\rho_\sigma(\mathbb{Z}^m)}\exp\left(-\frac{\|\mathbf{z}^* - \mathbf{Sc}\|^2}{2\sigma^2}\right) + \frac{1}{2\rho_\sigma(\mathbb{Z}^m)}\exp\left(-\frac{\|\mathbf{z}^* + \mathbf{Sc}\|^2}{2\sigma^2}\right) \\ &= \frac{1}{2\rho_\sigma(\mathbb{Z}^m)}\exp\left(-\frac{\|\mathbf{z}^*\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right)\left(e^{-\frac{\langle \mathbf{z}^*, \mathbf{Sc}\rangle}{\sigma^2}} + e^{\frac{\langle \mathbf{z}^*, \mathbf{Sc}\rangle}{\sigma^2}}\right) \\ &= \frac{1}{\rho_\sigma(\mathbb{Z}^m)}\exp\left(-\frac{\|\mathbf{z}^*\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle \mathbf{z}^*, \mathbf{Sc}\rangle}{\sigma^2}\right) .\end{aligned}$$

The desired output distribution is the centered Gaussian distribution $f(\mathbf{z}^*) = \rho_\sigma(\mathbf{z}^*)/\rho_\sigma(\mathbb{Z}^m)$. Thus, by Lemma 2.1, one should accept the sample $\mathbf{z}^*$ with probability:

$$p_{\mathbf{z}*} = \frac{f(\mathbf{z}^*)}{Mg_{\mathbf{Sc}}(\mathbf{z}^*)} = 1\Big/\left(M\exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle \mathbf{z}^*, \mathbf{Sc}\rangle}{\sigma^2}\right)\right),$$

where M is chosen large enough so that $p_{\mathbf{z}*} \leqslant 1$. Note that $\cosh(x) \geqslant 1$ for any x, so it suffices that

$$M = e^{\frac{1}{2\alpha^2}} \tag{2}$$

where α is such that $\sigma \geqslant \alpha \cdot \|\mathbf{Sc}\|$.

Bound on $\|\mathbf{Sc}\|$. Notice that if we fix the repetition rate M, then the standard deviation of the signature $\mathbf{z}$, and therefore also its size, only depend on the maximum possible norm of the vector $\mathbf{Sc}$. For this reason, it is important to obtain a bound as tight as possible on this product. Several upper bounds on $\|\mathbf{Sc}\|$ can be used such as $\|\mathbf{Sc}\| \leqslant \|\mathbf{c}\|_1 \cdot \|\mathbf{S}\| = \kappa \|\mathbf{S}\|$ (as in [24]) or $\|\mathbf{Sc}\| \leqslant s_1(\mathbf{S}) \cdot \|\mathbf{c}\| = s_1(\mathbf{S}) \cdot \sqrt{\kappa}$ where $s_1(\mathbf{S})$ is the singular norm of $\mathbf{S}$. Here we introduce a new measure of $\mathbf{S}$, adapted to the form of $\mathbf{c}$, which helps us achieve a tighter bound than with all previous methods. We believe that this norm and the technique for bounding it could be of independent interest.

Definition 3.1. *For any integer* κ*, we define* $N_\kappa \colon \mathbb{R}^{m \times n} \to \mathbb{R}$ *as:*

$$N_\kappa(\mathbf{X}) = \max_{\substack{I \subset \{1,\dots,n\} \\ \#I = \kappa}} \sum_{i \in I} \left(\max_{\substack{J \subset \{1,\dots,n\} \\ \#J = \kappa}} \sum_{j \in J} T_{i,j} \right) \quad \text{where } \mathbf{T} = \mathbf{X}^t \cdot \mathbf{X} \in \mathbb{R}^{n \times n} .$$

The following proposition states that $\sqrt{N_\kappa(\mathbf{S})}$ is also an upper bound for $\|\mathbf{Sc}\|$.

Proposition 3.2. *Let* $\mathbf{S} \in \mathbb{R}^{m \times n}$ *be a real matrix. For any* $\mathbf{c} \in \mathbb{B}^n_\kappa$*, we have* $\|\mathbf{Sc}\|^2 \leqslant N_\kappa(\mathbf{S})$.

In practice, we will use this upper bound to bound $\|\mathbf{Sc}\|$ and derive the parameters. Some secret keys $\mathbf{S}$ will be rejected according to the value of $N_\kappa(\mathbf{S})$, which is easily computable. In addition to the gain from the use of bimodal Gaussians, this new upper bound lowers the standard deviation σ by a factor $\approx \sqrt{\kappa}/2$ compared to [24].

3.3 Security of BLISS

Any existential forger against our signature scheme can solve the $\mathcal{R}\text{-SIS}^{\mathcal{K}}_{q,n,m,\beta}$ problem for $\beta = 2B_2$ where $\mathcal{K}$ is the distribution induced by the public-key generation algorithm.

Theorem 3.3. *Suppose there is a polynomial-time algorithm* $\mathcal{F}$ *which makes at most* s *queries to the signing oracle and* h *queries to the random oracle* H*, and succeeds in forging with non negligible probability* δ*. Then there exists a polynomial-time algorithm which can solve the* $\mathcal{R}\text{-SIS}^{\mathcal{K}}_{q,n,m,\beta}$ *problem for* $\beta = 2B_2$ *with probability* $\approx \frac{\delta^2}{2(h+s)}$*. Moreover the signing algorithm produces a signature with probability* $\approx 1/M$ *and the verifying algorithm accepts a signature produced by an honest signer with probability at least* $1 - 2^m$*.*

The proof of the theorem follows from standard arguments, and is simpler and tighter than the proof of [24]. In a nutshell, the fact that the distribution of the signatures in the scheme does not depend on the secret key means that the simulator can "sign" arbitrary messages without having the secret key by programming the random oracle. Then when the adversary produces a forgery, the simulator can extract a solution to the SIS problem. The proof is provided in the full version of this paper [5].

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th Annual ACM Symposium on Theory of Computing, Philadephia, Pennsylvania, USA, May 22–24, pp. 99–108. ACM Press (1996)
2. Bellare, M., Palacio, A.: GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002)
3. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011)
4. Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (2012)
5. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. Cryptology ePrint Archive (2013)
6. Ducas, L., Nguyen, P.Q.: Learning a zonotope and more: Cryptanalysis of ntrusign countermeasures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 433–450. Springer, Heidelberg (2012)
7. Fischer, J.-B., Stern, J.: An efficient pseudo-random generator provably as secure as syndrome decoding. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 245–255. Springer, Heidelberg (1996)
8. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008)
9. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
10. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st Annual ACM Symposium on Theory of Computing, Bethesda, Maryland, USA, May 31-June 2, pp. 169–178. ACM Press (2009)
11. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: In, R.E., Ladner, C. (eds.) 40th Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, pp. 197–206. ACM Press (2008)
12. Gentry, C., Szydlo, M.: Cryptanalysis of the revised NTRU signature scheme. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 299–320. Springer, Heidelberg (2002)
13. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997)
14. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: STOC, pp. 545–554 (2013)
15. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 530–547. Springer, Heidelberg (2012)
16. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Whyte, W.: Practical Lattice-Based Cryptography: NTRUEncrypt and NTRUSign. In: The LLL Algorithm: Survey and Applications. Information Security and Cryptography. Springer (2009)
17. Hoffstein, J., Pipher, J., Howgrave-Graham, N., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital signatures using the NTRU lattice. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 122–140. Springer, Heidelberg (2003)

18. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
19. Hoffstein, J., Pipher, J., Silverman, J.H.: NSS: An NTRU lattice-based signature scheme. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 211–228. Springer, Heidelberg (2001)
20. Howgrave-Graham, N.: A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 150–169. Springer, Heidelberg (2007)
21. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Karloff, H.J., Pitassi, T. (eds.) 44th Annual ACM Symposium on Theory of Computing, New York, NY, USA, May 19-22, pp. 1219–1234. ACM Press (2012)
22. Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 162–179. Springer, Heidelberg (2008)
23. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009)
24. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012)
25. Lyubashevsky, V., Micciancio, D.: Generalized compact Knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. Part II, LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
26. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010)
27. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
28. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007)
29. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. Journal of Cryptology 22(2), 139–160 (2009)
30. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
31. Rückert, M.: Lattice-based blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 413–430. Springer, Heidelberg (2010)
32. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011)
33. von Neumann, J.: Various techniques used in connection with random digits. J. Research Nat. Bur. Stand., Appl. Math. Series 12, 36–38 (1951)