Doctoral Thesis

Constructive and Destructive Aspects of Euclidean Lattices in Cryptography

Supervisor: Mehdi Tibouchi, Masayuki Abe

Department of Social Informatics
Graduate School of Informatics
Kyoto University
Japan

Chao Sun

Constructive and Destructive Aspects of Euclidean Lattices in Cryptography

Chao Sun

Abstract

Generally speaking, the field of cryptography consists of two aspects, namely *constructions* (the constructive aspect) and *cryptanalysis* (the destructive aspect). On the one hand, constructions aim to establish communication schemes that are both efficient and secure. On the other hand, the goal of cryptanalysis is to destruct schemes and recover information from the communication.

Nowadays, the widely deployed public key cryptographic schemes (such as RSA, DSA, ECDSA) are based on the hardness of integer factorization problem or the discrete logarithm problem. However, all of these problems can be efficiently solved on a quantum computer using Shor's algorithm. Even though current quantum computers are still not powerful enough to break any real cryptographic algorithms in practice, these schemes will be not secure anymore when large quantum computers are available in the future. Therefore, researchers are designing new cryptographic systems (so-called *post-quantum cryptography*) to prepare for the potential threat of quantum computers.

Among all the candidates in post-quantum cryptography, *lattice-based cryptography* is the most popular one, mainly because there are several advantages: from a algorithmical point of view, lattice-based schemes consist of linear operations on vectors and matrices, which makes it simple and parallelizable. In terms of security, lattice-based cryptography has strong security guarantee that *average-case* random instances are as hard as *worst-case* approximate variants of NP-hard problems. Besides, lattice-based cryptography supports versatile applications, including very advanced ones such as fully homomorphic encryption.

In this thesis, we study both aspects in lattice-based cryptography. In an intermediate aspect between constructive and destructive aspects (in the sense that security analysis can help determine paramters of constructions), we first study the *security of binary error LWE*. LWE is one of the central problems in lattice cryptography and binary error LWE is the particular case of LWE in which errors are chosen in $\{0, 1\}$.

It has various cryptographic applications, and in particular, has been used to construct efficient encryption schemes for use in constrained devices. We examine more generally how the hardness of binary error LWE varies with the number of available samples, using a simpler (but asymptotically equivalent) variant of the Gröbner basis algorithm. Besides, we generalize the uniform binary error LWE to the non-uniform case and analyze about the hardness of the non-uniform binary error LWE with respect to the error rate and the number of available samples.

In the destructive aspect, we study *lattice attacks on (EC)DSA*. Historically, lattice originally emerges as a powerful cryptanalysis tool to study the security of public key cryptography. Actually, *lattice reduction* has been used to attack (EC)DSA with partially known nonces. The attack itself has seen limited development and the lattice construction based on the signatures and known nonce bits remain the same. We propose a new idea to improve the attack: carry out an exhaustive search on some bits of the secret key. This turns the problem from a single bounded distance decoding (BDD) instance in a certain lattice to multiple BDD instances in a fixed lattice of larger volume but with the same bound. As a result, our analysis suggests that our technique is competitive or outperforms the state of the art for parameter ranges corresponding to the limit of what is achievable using lattice attacks so far. We also show that variants of this idea can also be applied to bits of the nonces or to filtering signature data. Besides, we use our technique to obtain an improved exploitation of the TPM–FAIL dataset.

In the constructive aspect, we study a very important primitive: *lattice-based signatures*. In particular, we introduce a novel trapdoor generation technique for Prest's hybrid sampler over NTRU lattices. Prest's sampler is used in particular in the Mitaka signature scheme (Eurocrypt 2022), a variant of the Falcon signature scheme, one of the candidates selected by NIST for standardization. Mitaka was introduced to address Falcon's main drawback, namely the fact that the lattice Gaussian sampler used in its signature generation is highly complex, difficult to implement correctly, to parallelize or protect against side-channels, and to instantiate over rings of dimension not a power of two to reach intermediate security levels. Prest's sampler is considerably simpler and solves these various issues, but the resulting scheme is still substantially less secure by Falcon and with much slower key generation. Our new trapdoor generation techniques solves all of those issues satisfactorily: it gives rise to a much simpler and faster key generation algorithm than Mitaka's (achieving similar speeds to Falcon), and is able to comfortably generate trapdoors reaching the same NIST security levels as Falcon as well. It can also be easily adapted to rings of intermediate dimensions, in order to support the same versatility as Mitaka in terms of parameter selection. All in all, this new technique combines all the advantages of both Falcon and Mitaka (and more) with none of the drawbacks.

Acknowledgements

First I would like to express my sincere thanks to my two supervisors, Mehdi Tibouchi and Masayuki Abe. Before coming to Kyoto University, I almost knew nothing about cryptography, but they led me into this field and taught me little by little. I gradually learnt a lot of things about cryptography and got attracted to this field. Without their help and support, it is impossible for me to finish this thesis. Even after working several years with them, I am still constantly surprised by their amazing intelligence, energy and friendliness.

I am grateful to my two advisors, Takayuki Kanda and Masaya Yasuda. They gave me valuable advice on my research topic, which guided me through finding new research ideas.

I would like to thank all the other members of Abe-Tibouchi Lab, who were really kind to me and helped me a lot whenever I had any trouble. It is really good experience to study together with them.

Besides, I receive a lot of input from my co-authors Thi Thu Quyen Nguyen, Thomas Espitau, Alexandre Wallet. I had helpful discussions and received comments from Ruosi Wan, Phong Nguyen. I would like to thank Thomas Espitau and Mehdi Tibouchi for hosting me at NTT. I would like to thank Yang Cao, Pierre Alain Fouque, Phong Nguyen and Kyosuke Yamashita for providing job information.

Last but not least, I want to thank my parents for their support and love.

Contents

Chapter 1

Introduction

The study of cryptography, like a *double-edged sword*, has two aspects: constructions (constructive aspect) and cryptanalysis (destructive aspect). Constructions aim to provide secure and efficient communication. By comparison, the goal of cryptanalysis is to recover information that is hidden in the communication. Despite the fact that cryptography has seen applications (e.g., Caesar cipher) more than two thousand years ago, for most of the time in history, it remained more like black art rather than science (even the word "cryptography" is relatively new).

1.1 Modern Public Key Cryptography

Most of the cryptography currently being used dates back to 1970s. In 1976, Diffie and Hellman published a paper called "new directions in cryptography" [DH76]. As the name suggests, they proposed a new idea of constructing cryptographic schemes based on the hardness of mathematical problems. However, Diffie and Hellman only gave the construction framework without proposing any concrete mathematical problems. Fortunately, only after two years of their publication, Rivest, Shamir and Adleman, proposed the famous RSA encryption scheme [RSA78], which is based on the hardness of integer factorization problem. Even up to now, most number-theoretic cryptography, still relies on the conjectured hardness of integer factorization or the discrete logarithm problem in certain groups. However, in 1994, Shor [Sho99] gave efficient quantum algorithms for all these problems, which would make number-theoretic systems insecure in the future when large-scale quantum computers are available. Therefore, researchers are designing the so-called post-quantum cryptography, i.e., candidates that are secure against quantum computers.

1.2 Cryptography in Social Informatics

In recent years, with the fast development of information science, there are a lot of applications of cryptography in social informatics. A few of them are listed below:

- Using digital signatures to authenticate. There are a lot of scenarios where we want to confirm that the received messages really come from the trusted party (not from some malicious bad adversaries). For example, suppose that the Nintendo company has published some video games and at some time, Nintendo might publish some patches that aim to fix some bugs in the previous versions. However, after downloading the patches from the internet, the users want to make sure that the downloaded patches really come from Nintendo, because the internet might be hijacked by the malicious guys. With digital signatures, nobody except the trusted party is able to publish a signature which matches the public key.

- Secure communication over the internet. Suppose that Alice wants to send some messages to Bob over the internet. Of course, Alice does not want anyone over the public channel to know her messages to Bob. Therefore, encryption schemes could be used to encrypt the plaintext messages into ciphertext. Besides, with digital signatures, Alice can also make sure that her messages are not modified over the public channels.

- Electronic money and cryptocurrency. In recent years, considerable interest has been found in electronic money (e.g., Paypay, Alipay, Linepay) and cryptocurrency (e.g., Bitcoin, ETH, USDT), which make the offline payments very convenient. Cryptography plays a central role in making those payment methods secure. When users pay their money at some stores, from the cryptographic perspective, essentially the users are issuing a digital signature with the private signing key that authenticates the transactions.

Still, there are many other applications of cryptography in social informatics. Therefore, it is important to study cryptogaphy in order to build a more secure informatic society.

1.3 Lattice-based Cryptography

Slightly informally, lattice is a pattern of grid that appears in the vector space $\mathbb{R}^n$. A lattice has a basis, which consists of a finite number of linearly independent vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n$. As shown in figure 1.1, this is a 2-dimensional lattice with basis $\mathbf{b}_1, \mathbf{b}_2$.

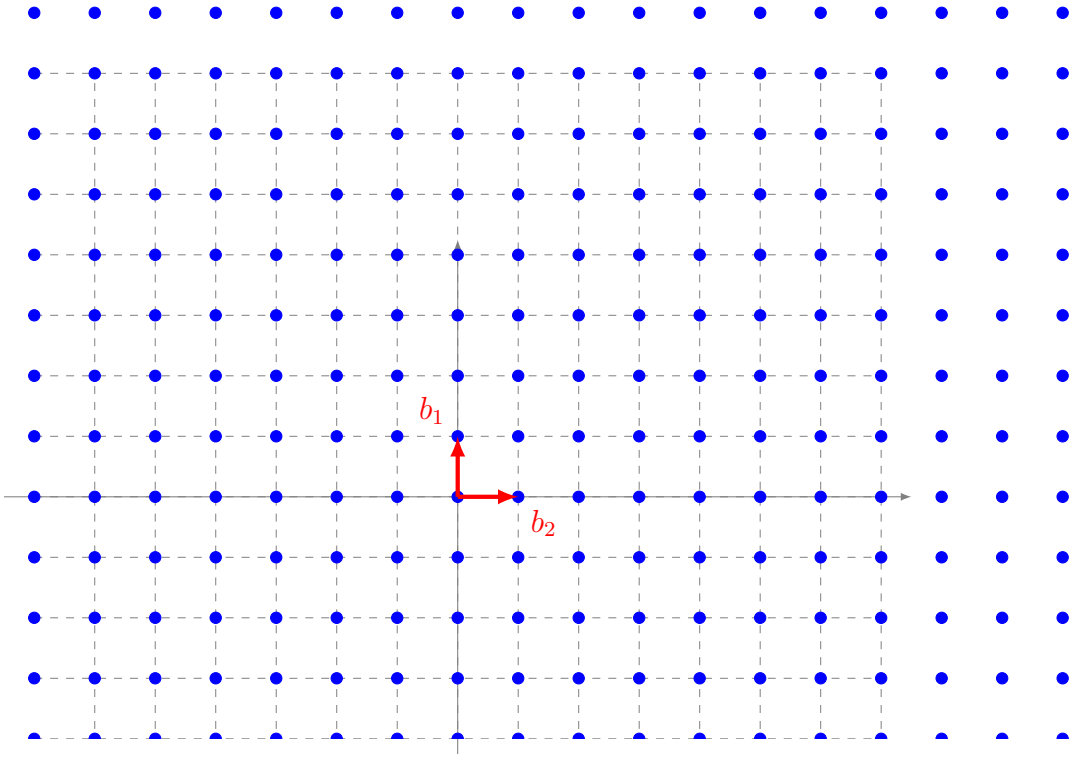


Figure 1.1: Lattices

1.3.1 Constructive Aspect of Lattice-based Cryptography

In the constructive aspect, lattice-based cryptography has developed for nearly 30 years.

Short integer solution problem: In 1996, Ajtai [Ajt96] introduced the short integer solutions (SIS) problem and proved that solving *average-case* SIS is at least as hard as approximating *worst-case* lattice problems. It turns out that SIS is extremely useful in constructing collision-resistant hash functions, one-way functions and digital signatures.

Learning with errors: In 2005, Regev introduced the learning with errors problem (LWE) [Reg10], which is one of the central problems in lattice-based cryptography. *Average-case* LWE, for suitable parameters, is as hard as worst-case lattice problems, and it is therefore very convenient to build secure lattice-based cryptographic schemes: it has been used to build various primitives, especially encryption schemes.

NTRU: In 1998, Hoffstein, Pipher and Silverman introduced the public-key encryption scheme NTRU [HPS98] (known as NTRUEncrypt in order to be distinguishable from NTRUSign). A few years later, they also proposed a digital signature scheme called NTRUSign. Although originally presented in a purely algebraic manner, it can be reformulated in terms of lattices to make everything neat and elegant. The NTRU cryptosystem is extremely efficient and has compact keys, thanks to the algebraic structures.

From GGH to GPV: In 1997, Goldreich, Goldwasser, and Halevi (GGH) [GGH97] proposed a public-key encryption scheme and a digital signature scheme. The main idea behind the GGH signature scheme is that a public key is some "bad" basis of some lattice, while the secret key is a "good" basis of the same lattice, which consists of relatively short and close to orthogonal vectors. In the GGH signature, a message is mapped a point h in the vector space $\mathbb{R}^n$. To sign a message, the "good" basis is used to find a lattice point near the message point h and the close lattice point is the signature. To verify the signature, one can just check the signature is a lattice point by using the "bad" basis and that the message is close to the signature.

The GGH scheme, as well as several successive variants of NTRUSign, were eventually broken by statistical attacks [GS02, NR06, DN12]: it turned out that signatures would reveal partial information about the secret trapdoor, that could then be progressively recovered by an attacker. This problem was finally solved in 2008, when Gentry, Peikert and Vaikuntanathan (GPV) [GPV08] showed how to use Gaussian sampling in the lattice in order to guarantee that signatures would reveal no information about the trapdoor.

Advantage of lattice-based cryptography: Interestingly, no efficient quantum algorithms are known for the problems typically used in lattice cryptography, which makes lattice cryptography a very promising candidate for post-quantum cryptography. Actually, NIST (National Institute of Standards and Technology, U.S.) has selected CRYSTALS-KYBER, CRYSTALS-Dilithium, FALCON and SPHINCS$^+$ as the PQC standardization. Of these candidates, 75% are lattice-based, mainly because lattice cryptography has the following advantages:

Efficient and easy to implement: Lattice-based cryptosystems are often simple and easy to implement. Consisting mainly of linear operations on vectors and matrices, lattice cryptography has fast speed. Moreover, constructions on some specific algebraic lattices over certain rings are very efficient. For instance, the NTRU [HPS98] system can be especially efficient, and in some cases even outperform the traditional cryptosystems.

Strong security guarantee: Cryptography inherently requires average-case intractability, i.e., problems for random instances are hard to solve. This is quite dif-

ferent from the worst-case notation of hardness usually considered in the theory of algorithms and NP-completeness, where a problem is considered hard if there merely exist some intractable instances. Problems that appear hard in the worst case often turn out to be easier on average. In a seminal work, Ajtai [Ajt96] gave a connection between the worst case and the average case for lattices: he proved that certain problems are hard on the average, as long as some related lattice problems are hard in the worst case. This result is quite meaningful in the sense that one can design cryptographic constructions and prove that they are infeasible to break, unless all instances of certain lattice problems are easy to solve.

Versatile applications: From lattices, it is possible to construct almost all the cryptographic primitives, including advanced ones such as fully homomorphic encryption, attribute based encryption.

1.3.2 Destructive Aspect of Lattice-based Cryptography

Somewhat surprisingly, due to historical reasons, lattices first appear as a cryptanalytic (destructive aspect) tool. A lattice has a infinite number of bases, and the goal of *lattice reduction*, is to find useful bases that are relatively short and close to orthogonal. From a mathematical point of view, lattice reduction has a long history, which dates back to the reduction theory of quadratic forms developed by Gauss, Lagrange and Hermite, and to Minkowski's geometry of numbers.

LLL and cryptanalysis: Although lattice reduction has a long history, however, it was not until 1982 that Lenstra, Lenstra and Lov*á*z invented a polynomial-time lattice reduction algorithm [LLL$^+$82], where they applied it to factor fractional polynomials. Subsequently, researchers immediately noticed the relation between lattice reduction and cryptography. It was used to break cryptosystems that are based on the knapsack problem [BO88, Sha82]. Interestingly, in 1996, Coppersmith observed the relation between lattices and polynomials. In an elegant work, he showed that lattice reduction can be used to find small solutions of polynomials [Cop97]. In particular, this led to attacks on RSA with specific parameters.

A few years later, Howgrave-Graham and Smart [HGS01], and later Shparlinski and Nguyen [NS02], found that attacking (EC)DSA if some bits of the nonces are known, can be solved via lattice reduction. However, when nonce leakage is very small, the attack becomes much more difficult. In 2013, with BKZ 2.0, Liu and Nguyen [LN13] were able to attack 160-bit DSA with 2-bit nonce leakage using the BKZ 2.0 algorithm introduced just a few years earlier [CN11], relying on a very high block size of 90, with pruned enumeration as the SVP oracle. In a very recent work [AH21b], Albrecht and Heninger utilize the state-of-the-art lattice reduction algorithm G6K [ADH$^+$19] together with the novel idea of predicate sieving to break

new records.

1.4 Contributions Overview

In this thesis, we study lattice-based cryptography from both aspects. Firstly, we study and analyze the security of binary error LWE. This work lies in between the constructive and destructive aspect, because the result can help set parameters of cryptographic schemes. Secondly, from a purely destructive aspect, we study and improve lattice attacks on (EC)DSA with nonce leakage. Thirdly, from a constructive aspect, we study lattice-based signatures and improve the Mitakasignature scheme to make it more secure and efficient.

1.4.1 Security Analysis of Binary Error LWE

For efficiency reasons, constructions often rely on variants of LWE (such as its ring version Ring-LWE [LPR10]) or instantiations in more aggressive ranges of parameters than those for which Regev's reduction to worst-case lattice problems holds. An important example is binary error LWE, where the error term is sampled from $\{0, 1\}$ (instead of from a wider discrete Gaussian distribution). Binary error LWE is a particularly simple problem with various interesting cryptographic applications, such as Buchmann et al.'s efficient lattice-based encryption scheme for IoT and lightweight devices [BGG+16] (based on the ring version of binary error LWE, with the additional constraint that the secret is binary as well).

However, the problem is not hard given arbitrarily many samples: in fact, an algebraic attack due to Arora and Ge [AG11] solves uniform Binary-Error LWE in polynomial time given around $n^2/2$ samples. The same approach can also be combined by Gröbner basis techniques to reduce the number of required samples [APS15]. On the other hand, Micciancio and Peikert [MP13] showed the Binary error LWE problem reduces to standard LWE (and thus is believed to be exponentially hard) when the number of samples is restricted to $n + O(n/\log n)$. Thus, the hardness of Binary error LWE crucially depends on the number of samples released to the adversary.

We show that a simple extension of the Arora-Ge attack (based on similar ideas as the Gröbner basis approach, but simpler and at least as fast) provides a smooth time-sample trade-off for binary error LWE: the attack can tackle any number of samples, with increasing complexity as the number of samples decreases. In particular, for binary error LWE with $\epsilon \cdot n^2$ samples ($\epsilon > 0$ is a constant), we obtain an attack in polynomial time $n^{O(1/\epsilon)}$, assuming standard heuristics on the polynomial system arising from the Arora-Ge approach. Similarly, for $n^{1+\alpha}$ samples ($\alpha \in (0, 1)$ constant), we obtain an attack in subexponential time $2^{\tilde{O}(n^{1-\alpha})}$. The precise complexity

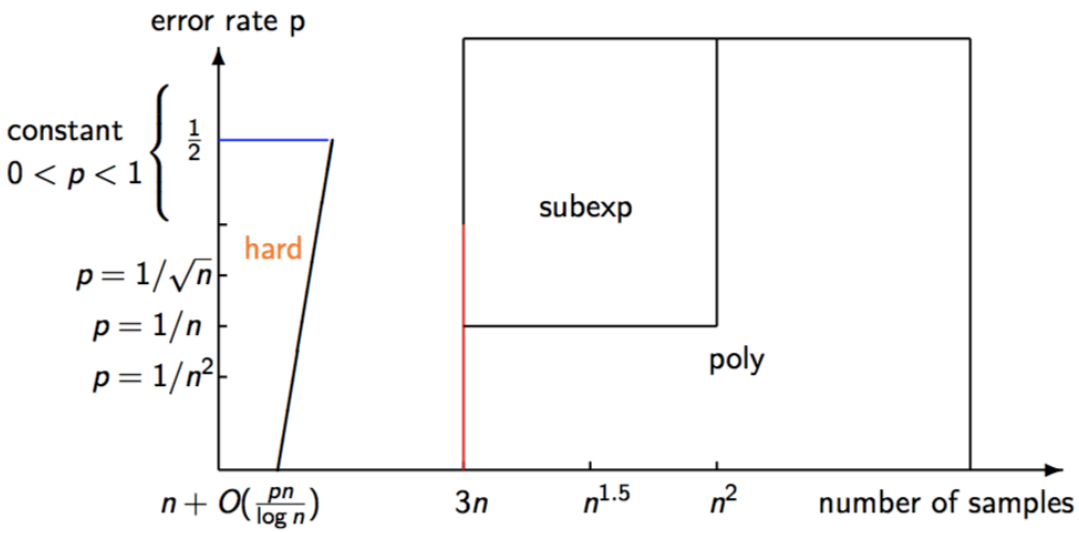


Figure 1.2: Hardness Result for Non-uniform Binary Error LWE

for any concrete number of samples is also easy to compute, which makes it possible to precisely set parameters for cryptographic schemes based on binary error LWE.

Besides, we make a generalization of the uniform binary error LWE to the non-uniform case, in which the error is chosen from $\{0, 1\}$ and the error is 1 with some probability p (and 0 with probability $1 - p$). We analyze this problem from two perspectives. As we can see in the figure 1.2, on the one hand, we show that for any error rate p, non-uniform binary error LWE is as hard as worst-case lattice problems as long as the number of samples is restricted. This is a generalization of the hardness proof given by Micciancio and Peikert to the non-uniform case. On the other hand, we show that when the error rate is $p = 1/n^{\alpha}(\alpha \geq 1)$, it can be solved in polynomial time with $O(n)$ samples, and when the error rate is $p = 1/n^{\alpha}$ $(0 < \alpha < 1)$, it can be solved in subexponential time with $O(n)$ samples.

1.4.2 Improving Lattice Attacks on (EC)DSA

In 1991, NIST (National Institute of Standards and Technology, U.S.) proposed DSA for use in their Digital Signature Standard (DSS). DSA is a variant of the Schnorr and ElGamal signature schemes (due to patent reasons), and ECDSA is the analogue in the context of elliptic curves. In the signing algorithm of (EC)DSA, there is a randomness k, which we usually call the *nonce*. If the same nonce k is used twice, due to the linear relation between the nonces and secret key, we can directly recover the secret key. Moreover, partial information about the nonces can lead to recovery of the full private key. As long as there are enough number of signatures provided, we

can attack biased nonce (EC)DSA by lattice reduction or Fourier analysis techniques [Ble00, AFG+14b, DHMP13, TTA18, ANT+20].

In the process of lattice attacks on (EC)DSA, we either get the full signing key or get nothing. Inspired by this, we propose a new idea that by guessing bits of the signing key or the nonces, we are able to construct a new lattice where lattice reduction is significantly easier. This new approach has a lot of advantages. Firstly, it is very easy to parallelize and simulate in the sense that we can assume that we have guessed the correct bits, thus avoiding a huge amount of computation. Secondly, it allows us to use batch-SVP and CVP with preprocessing techniques, which can further improve the computation cost. Finally, it is compatible with all the existing techniques [LN13, AH21b, JSSS20].

As additional contributions, we also show that the same idea can be applied to filtering some of the signatures to construct lattices that are easier to attack (resulting in a data-time trade-off reminiscent to what can be achieved in Bleichenbacher's attack). Furthermore, we carry out experiments on the TPM-FAIL dataset [MSEH20] and apply our techniques to key recovery. While the original attack requires about 40000 signatures, with the method of guessing bits, we are able to recover the secret key with only around 800 signatures, which is comparable to the results achieved in Minerva [JSSS20].

1.4.3 Constructing Efficient and Secure Lattice-based Signatures

Despite the fact that GPV-framework successfully hides information of the trapdoor basis, the resulting signature schemes is not efficient from two aspects: on the one hand, to reach a standard level of security, the public basis matrix of the lattice has to be set as a $m \times n$ matrix, where $n = 512, m \approx 24n$. For these typical parameters, the size of public key and signature is 38 kB, 26 kB respectively [CGM19], which is still quite large compared with traditional digital signature. On the other hand, during the signing phase, the Klein-GPV sampler (a randomized variant of Babai's nearest plane algorithm [Bab86]) is used, whose time complexity is quadratic in the lattice dimension n. This makes the signing phase rather inefficient.

In order to deal with the first issue, in 2014, Ducas, Lyubashevsky and Prest (DLP) instantiated the GPV framework over NTRU lattices. They carefully analyze the Euclidean length of the Gram-Schmidt orthogonalization of NTRU lattices and are able to generate the trapdoor basis from a distribution where the Gram-Schmidt norm satisfies a certain quality condition. As a result, the DLP signature scheme has rather compact public key size and signature size (less than 1kB).

To tackle the second issue, Ducas and Prest proposed the FFO sampling algorithm whose asymptotic time complexity is quasilinear in the lattice dimension. Actually, the NIST postquantum standardization candidate Falcon [PFH+22] is essentially a combination of the DLP signature and FFO sampler. However, the FFO sampler has

several drawbacks: it is a very complicated algorithm that is very difficult to implement correctly, parallelize or protect against side-channel attacks. At Eurocrypt 2022, Espitau et al. proposed the Mitaka signature scheme, which is a parallelizable, simpler, maskable variant of Falcon. In Mitaka, the FFO sampler is replaced with Prest's hybrid sampler [Pre15]. The main contribution of Mitaka is an improved trapdoor generation technique that improved the security level of the resulting signature scheme. However, Mitaka is less secure than Falcon in equal dimension (over 20 bits over lattices of dimension 512, and more than 50 bits over lattices of dimension 1024), with a much slower and more contrived key generation algorithm as well.

We introduce a novel trapdoor generation technique for Prest's hybrid sampler that solves the issues faced by Mitaka in a natural and elegant fashion. Our technique gives rise to a much simpler and faster key generation algorithm than Mitaka's (achieving similar speeds to Falcon), and it is able to comfortably generate trapdoors reaching the same NIST security levels as Falcon. It can also be easily adapted to rings of intermediate dimensions, in order to support the same versatility as Mitaka in terms of parameter selection (just with better security). All in all, this new technique achieves in some sense the best of both worlds between Falcon and Mitaka.

1.4.4 Contributions to Social Informatics

Since the potential advent of large-scale quantum computers, it is important to move to postquantum cryptography. The security analysis of binary error LWE can be quite useful to construct cryptosystems that are based on LWE, especially for lightweight devices, where implementation of cryptosystems based on standard LWE is rather inefficient. In particular, our work sheds light on the security of the IoT-friendly scheme of Buchmann et al [BGG$^+$16].

The improved lattice attacks on (EC)DSA is very important to enhance the security for applications where (EC)DSA is used. For example, in bitcoin transactions, we should be extremely careful that the random number generated does not have any bias, which might lead to the full private signing key recovery. Actually, in [BH19], Breitner and Heninger analyze a lot of bitcoin transactions on the blockchain and find that many of them are insecure which essentially leads to key recovery. Due to legal issues, they only check that the private key can be recovered but do not steal any money. What's more, implementation is algorithmically correct does not necessarily mean that it is secure for sure. In a paper at CHES 2020 [JSSS20], the authors analyze a lot of cryptographic libraries and find that many of them are insecure, leaking side-channel information about the nonce. Therefore, in order to avoid nonce leakage, implementation should be carefully done to protect side-channel attacks.

Since in the future, current cryptosystems will be replaced with postquantum cryptosystems, the work of constructing efficient and secure lattice-based signatures is also very meaningful in the sense that our signature may serve as a potential candi-

date as postquantum digital signatures. With these signatures, we could even protect cryptosystems from being broken by quantum computers. In fact, our work provides an attractive alternative to NIST standard Falcon that is much easier to implement correctly and more suitable on constrained devices.

1.5 Thesis Outline

In Chapter 2, we study the security of binary error LWE. In chapter 3, we present our improved attacks on (EC)DSA. In chapter 4, we show how to construct efficient and secure lattice-based signatures. In chapter 5, we give a summary of all the contributions.

Chapter 2

Security Analysis of Binary Error LWE

2.1 Learning with Errors

The LWE problem asks to recover a secret $\mathbf{s} \in \mathbb{Z}_q^n$, given a system of linear approximate equations. For example, an instance of LWE [Reg10] could be:

$$\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &\approx 8 \pmod{17} \\
13s_1 + 14s_2 + 14s_3 + 6s_4 &\approx 16 \pmod{17} \\
6s_1 + 10s_2 + 13s_3 + s_4 &\approx 3 \pmod{17} \\
10s_1 + 4s_2 + 12s_3 + 16s_4 &\approx 12 \pmod{17} \\
9s_1 + 5s_2 + 9s_3 + 6s_4 &\approx 9 \pmod{17} \\
3s_1 + 6s_2 + 4s_3 + 5s_4 &\approx 16 \pmod{17}
\end{aligned}$$

Each equation is satisfied up to some small error, sampled independently according to some known distribution (typically a discrete Gaussian distribution). The goal is to recover the secret $\mathbf{s}$. If the equation held without error, finding $\mathbf{s}$ would simply amount to solving a system of linear equations. We could therefore recover the secret $\mathbf{s}$ in polynomial time $O(n^\omega)$, where $2 \leq \omega \leq 3$ is the complexity exponent of linear algebra ($\omega \approx 2.37$ with the best known approach [LG14]). However, the errors introduced in LWE typically make the problem much harder. Formally, the LWE problem can be defined as follows.

Definition 2.1.1 (LWE)**.** The (search) LWE problem, defined with respect to a dimension n, a modulus q and an error distribution χ over $\mathbb{Z}_q$, asks to recover a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ given polynomially many samples of the form

$$\big(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \tag{2.1}$$

where **a** is uniformly random in $\mathbb{Z}_q^n$, and e is sampled according to χ. One can optionally specify the number of available samples as an additional parameter.

Remark. One can also similarly define a decision variant of the LWE problem, which asks to distinguish the distribution of the samples (2.1) above from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. The LWE problem given m samples has a simple expression in matrix form: it asks to recover **s** from the pair $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is a uniformly random matrix, and $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q$, where all the coefficients of $\mathbf{e} \in \mathbb{Z}_q^m$ are sampled independently from χ.

2.2 Binary Error LWE

The binary error LWE is simply the special case of Definition 2.1.1 where χ is the uniform distribution over $\{0, 1\}$. In other words:

Definition 2.2.1 (Binary Error LWE)**.** The binary error LWE with parameters n, m and q asks to recover the vector $\mathbf{s} \in Z_q^n$ from m samples of the form:

$$\big(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

where **a** is uniformly random in $\mathbb{Z}_q^n$, and e is uniform in $\{0, 1\}$.

The dimension n is the main security parameter, and both m and q are typically chosen as polynomially bounded functions of n. In this thesis, we assume that $q = n^{\Theta(1)}$.

Non-uniform binary error LWE is simply the special case of Definition 2.1.1 where χ is the non-uniform distribution over $\{0, 1\}$. In other words:

Definition 2.2.2 (Non-uniform Binary Error LWE)**.** Let $\mathcal{B}$ be a distribution over $\{0, 1\}$ that samples 1 with probability p and 0 with probability $1 - p (0 < p < 1)$, The non-uniform binary error LWE with parameters n, m and q asks to recover the vector $\mathbf{s} \in \mathbb{Z}_q^n$ from m samples of the form:

$$\big(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

where **a** is uniformly random in $\mathbb{Z}_q^n$, and e is sampled according to $\mathcal{B}$.

The dimension n is the main security parameter, and both m and q are typically chosen as polynomially bounded functions of n.

Uniqueness of Solutions for LWE

Theorem 1. *Suppose that the following condition is satisfied:*

$$m \geq n \cdot \left(1 + \frac{c}{\log q}\right)$$

for some $c > \log 3$*. Then, the binary error LWE problem with parameters* n, m, q *has a unique solution with overwhelming probability.*

Proof. Indeed, suppose that two solutions $\mathbf{s} \neq \mathbf{s}'$ exist to the binary error LWE challenge $(\mathbf{A}, \mathbf{b})$. This means that there exists binary error vectors $\mathbf{e}, \mathbf{e}'$ such that:

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{A}\mathbf{s}' + \mathbf{e}'.$$

As a result, the vector $\mathbf{t} = \mathbf{s}' - \mathbf{s} \neq 0$ satisfies $\mathbf{A}\mathbf{t} = \mathbf{e} - \mathbf{e}' \in \{-1, 0, 1\}^m$. It thus suffices to prove that for a random $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, such a vector $\mathbf{t}$ can only exist with negligible probability.

We can proceed as follows: fix $\mathbf{t} \in \mathbb{Z}_q^n \setminus \{0\}$. For a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the probability that $\mathbf{A}\mathbf{t} \in \{-1, 0, 1\}^m$ is exactly $3^m/q^m$, since the product vector is uniformly distributed in $\mathbb{Z}_q^m$. As a result, the union bound shows that:

$$\Pr_{\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}} \left[\exists \mathbf{t} \in \mathbb{Z}_q^m \setminus \{0\}, \mathbf{A}\mathbf{t} \in \{-1, 0, 1\}^m\right] \leq \left(\frac{3}{q}\right)^m \cdot q^n$$

since there are fewer than q^n possible vectors $\mathbf{t}$.

Therefore, assuming without loss of generality that $q > 3$, the probability ϵ that the challenge has at least two solutions is bounded as:

$$\begin{aligned}
\epsilon &\leq \left(\frac{3}{q}\right)^m \cdot q^n \\
\log \epsilon &\leq m \log\left(\frac{3}{q}\right) + n \log q \\
&\leq n\left(1 + \frac{c}{\log q}\right) \log\left(\frac{3}{q}\right) + n \log q \\
&= n\left(\log 3 - \log q + \frac{c \log 3}{\log q} - c + \log q\right) \\
&= n\big(\log 3 - c + o(1)\big)
\end{aligned}$$

and since $c > \log 3$, it follows that ϵ is negligible. □

2.3 Mathematical Background

In this section, we introduce some mathematical background, which are extremely useful in crypto-analysis.

2.3.1 Cauchy Integral Formula

Theorem 2 (Cauchy)**.** *Let C be a simple closed curve in the complex plane and f a holomorphic function on a region containing C and its interior. Assume C is oriented counterclockwise. Then for any z_0 inside C:*

$$f(z_0) = \frac{1}{2\pi i} \oint_C \frac{f(z)}{z - z_0}\, dz.$$

Theorem 3 (Cauchy for derivatives)**.** *Under the same hypotheses, we have for all $n \geq 0$:*

$$f^{(n)}(z_0) = \frac{n!}{2\pi i} \oint_C \frac{f(z)}{(z - z_0)^{n+1}}\, dz.$$

2.3.2 Laplace's method

Theorem 4. *Let $\Phi : [a, b] \to \mathbb{R}$, $\psi : [a, b] \to \mathbb{C}$ be smooth functions. We assume that $\Phi'' > 0$ over $[a, b]$ and there exists $x_0 \in (a, b)$ such that $\Phi'(x_0) = 0$. Then, the following asymptotic estimate holds for $s \to +\infty$:*

$$\int_a^b e^{-s\Phi(x)} \phi(x)\, dx = e^{-s\Phi(x_0)} \left[\frac{A}{\sqrt{s}} + O\Big(\frac{1}{s}\Big) \right]$$

where $A = \psi(x_0)\sqrt{2\pi/\Phi''(x_0)}$.

2.3.3 Standard Tail Bound

There are some standard results in probability theory that are often used in the field of cryptography. In this section, we recall these useful results:

Markov's Inequality

Lemma 5. Let X be a non-negative random variable and $v > 0$, Then:

$$Pr[X \geq v] \leq Exp[X]/v.$$

Proof. We have

$$\begin{aligned} Exp[X] &= \sum_{x \geq 0} Pr[X = x] \cdot x \\ &\geq \sum_{0 \leq x < v} Pr[X = x] \cdot 0 + \sum_{x \geq v} Pr[X = x] \cdot v \\ &= Pr[X \geq v] \cdot v \end{aligned}$$

□

Markov's inequality is used when only the expectation of the random variable X is known. In some sense, this bound is a bit loose. If the variance of X is known, some better bounds exist.

Chebyshev's Inequality

Lemma 6. Let X be a random variable and $\delta > 0$. Then

$$Pr[|X - Exp[X]| \geq \delta] \leq \frac{Var[X]}{\delta^2}$$

Proof. Define the non-negative random variable $Y = (X - Exp[X])^2$, and apply Markov's inequality:

$$\begin{aligned} \Pr[|X - \text{Exp}[X]| \geq \delta] &= \Pr\left[(X - \text{Exp}[X])^2 \geq \delta^2\right] \\ &\leq \frac{\text{Exp}\left[(X - \text{Exp}[X])^2\right]}{\delta^2} \\ &= \frac{\text{Var}(X)}{\delta^2} \end{aligned}$$

□

Chernoff Bound

Theorem 7. *Let $X = \sum_{i=1}^{n} X_i$, where $X_i = 1$ with probability p_i and $X_i = 0$ with probability $1 - p_i$, and all X_i are independent. Let $\mu = \mathbb{E}(X) = \sum_{i=1}^{n} p_i$. Then*

- *Upper Tail:* $\mathbb{P}(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2}{2+\delta}\mu}$ *for all* $\delta > 0$.
- *Lower Tail:* $\mathbb{P}(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2}$ *for all* $0 < \delta < 1$.

In order to prove this theorem, we need some additional lemmas.

Lemma 8. If $X = \sum_{i=1}^{n} X_i$ where $X_1, X_2 \cdots, X_n$ are independent random variables, then

$$M_X(s) = \prod_{i=1}^{n} M_{X_i}(s)$$

where $M_X(s) = \mathbb{E}\left(e^{sX}\right)$.

Proof. For any $s > 0$

$$\begin{aligned} M_X(s) &= \mathbb{E}\left(e^{sX}\right) = \mathbb{E}\left(e^{s\sum_{i=1}^{n} X_i}\right) \\ &= \mathbb{E}\left(\prod_{i=1}^{n} e^{sX_i}\right) \\ &= \prod_{i=1}^{n} \mathbb{E}\left(e^{sX_i}\right) \text{ (by independence)} \\ &= \prod_{i=1}^{n} M_{X_i}(s) \end{aligned}$$

This lemma allows us to prove a Chernoff bound by bounding the moment generating function of each X_i individually. □

Lemma 9. Let Y be a random variable that takes value 1 with probability p and 0 with probability $1 - p$. Then, for all $s \in \mathbb{R}$:

$$M_Y(s) = \mathbb{E}\left(e^{sY}\right) \leq e^{p(e^s - 1)}$$

Proof.

$$\begin{aligned} M_Y(s) &= \mathbb{E}\left(e^{sY}\right) \\ &= p \cdot e^s + (1 - p) \cdot 1 \\ &= 1 + p\left(e^s - 1\right) \\ &\leq e^{p(e^s - 1)} \end{aligned}$$

□

Chernoff bound can be proved with the above two lemmas.

Proof. Applying the above two lemmas, we obtain

$$M_X(s) \leq \prod_{i=1}^{n} e^{p_i(e^s - 1)} = e^{(e^s - 1)\sum_{i=1}^{n} p_i} \leq e^{(e^s - 1)\mu}$$

using that $\sum_{i=1}^{n} p_i = \mathbb{E}(X) = \mu$. For the proof of the upper tail, for any $s > 0$,

$$\begin{aligned} \mathbb{P}(X \geq a) &= \mathbb{P}\left(e^{sX} \geq e^{sa}\right) \\ &\leq \frac{\mathbb{E}\left(e^{sX}\right)}{e^{sa}} \end{aligned}$$

by Markov's inequality. Setting $a = (1+\delta)\mu$ and $s = \ln(1+\delta)$, we have

$$\begin{aligned}\mathbb{P}(X \geq (1+\delta)\mu) &\leq e^{-s(1+\delta)\mu} e^{(e^s-1)\mu} \\ &= \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu\end{aligned}$$

Using the following inequality for $x > 0$

$$\ln(1+x) \geq \frac{x}{1+x/2}$$

We obtain

$$\mu(\delta - (1+\delta)\ln(1+\delta)) \leq -\frac{\delta^2}{2+\delta}\mu$$

Hence, we have the desired bound for the upper tail:

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu e^{-\frac{\delta^2}{2+\delta}\mu}$$

□

The proof of the lower tail bound is quite similar, which we omit here.

Hoeffding's Inequality

According to [Wik22], let $X_1, \cdots, X_n$ be indepedent random variables such that $\forall i$, $a \leq X_i \leq b$, and let $S_n = X_1 + \cdots + X_n$. Then Hoeffding's inequality says that $\forall$ $t > 0$,

$$\begin{aligned}\mathrm{P}(S_n - E(S_n) \geq t) &\leq exp\big(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\big) \\ \mathrm{P}|(S_n - E(S_n)| \geq t) &\leq 2exp\big(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\big)\end{aligned}$$

where $E(S_n)$ is the expectation of S_n. In particular, we are mostly interested in the important special case of identically distributed Bernoulli random variables: $X_i = 1$ with probability p and $X_i = 0$ with probability $1-p$ ($i = 1, \cdots, n$). And again let $S_n = X_1 + \cdots + X_n$. The probability that $S_n \leq k$ can be exactly quantified by the following expression:

$$\mathrm{P}(S_n \leq k) = \sum_{i=0}^{k} \left(\begin{array}{c} n \\ i \end{array} \right) p^i (1-p)^{n-i}$$

For this special case, Hoeffding's inequality states that

$$\mathrm{P}((p-\varepsilon)n \leq S_n \leq (p+\varepsilon)n) \geq 1 - 2\exp\left(-2\varepsilon^2 n\right)$$

2.3.4 Gussian Distribution

A random variable X is said to be normally distributed with mean μ and variance δ^2 if its probability density function is

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad -\infty < x < \infty$$

The Gaussian distribution is usually represented as

$$X \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

and the graph of the Gaussian distribution is a bell-shaped curve that is symmetric about the mean μ. However, this is a univariate Gaussian distribution.

Discrete Gaussian Distribution: In the field of lattice cryptography, a multi-dimensional Gaussian distribution is often used. The definition follows:

Definition 2.3.1. For any positive integer n and real number $s > 0$, which is taken to be $s = 1$ when omitted, define the Gaussian function $\rho_s : \mathbb{R} \to \mathbb{R}^+$ of parameter s as

$$\rho_s(\mathbf{x}) := \exp\left(-\pi\|\mathbf{x}\|^2/s^2\right) = \rho(\mathbf{x}/s)$$

Notice that ρ_s is invariant under rotations of $\mathbb{R}^n$ and that $\rho_s(\mathbf{x}) = \prod_{i=1}^n \rho_s(x_i)$. The continuous Gaussian distribution D_s of parameter s over $\mathbb{R}^n$ is defined to have probability density function proportional to ρ_s, i.e.,

$$f(\mathbf{x}) := \rho_s(\mathbf{x}) / \int_{\mathbb{R}^n} \rho_s(\mathbf{z}) d\mathbf{z} = \rho_s(\mathbf{x})/s^n$$

For a lattice coset $\mathbf{c} + \mathcal{L} \subset \mathbb{R}^n$ and parameter $s > 0$, the discrete Gaussian probability distribution $D_{\mathbf{c}+\mathcal{L},s}$ is simply the Guassian distribution restricted to the coset.

$$D_{\mathbf{c}+\mathcal{L},s}(\mathbf{x}) \propto \begin{cases} \rho_s(\mathbf{x}) & \text{if } \mathbf{x} \in \mathbf{c} + \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

In cryptography analysis, the following tail bound of Gaussian distribution is often used:

Theorem 10. *[APS15] Let χ denote the Gaussian distribution with standard deviation δ and mean zero. Then, for all C>0, it holds that:*

$$\Pr\left[e \leftarrow_{\$} \chi : |e| > C \cdot \sigma\right] \leq \frac{2}{C\sqrt{2\pi}} \exp\left(-C^2/2\right)$$

Proof.

$$
\begin{aligned}
&\Pr[e \leftarrow \sigma\chi : |e| > C \cdot \sigma] \\
&= 2 \cdot \int_{C\cdot\sigma}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\
&= \frac{2}{\sqrt{2\pi}} \int_{C\cdot\sigma}^{\infty} \frac{1}{\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\
&\leq \frac{2}{\sqrt{2\pi}} \int_{C\cdot\sigma}^{\infty} \frac{t}{C\sigma^2} \exp\left(-\frac{t^2}{2\sigma^2}\right) dt \\
&= \frac{2}{C\sqrt{2\pi}} \exp\left(-C^2/2\right)
\end{aligned}
$$

□

2.4 Algorithms for Attacking LWE

In this section, we discuss some algorithms that are used to attack LWE. Basically, these algorithms can be divided into the following groups:

- Naive algorithm. Since the secret **s** is a n-dimensional vector, an exhaustive search would be trying all the possible secret **s** and see whether the computed error is small.
- Combinatorial algorithm. An interesting algorithm follows from the work of Blum, Kalai, and Wasserman [BKW03]. It is based on an idea that allows to find a small set S of equations among the $2^{O(n)}$ equations. By summing up these equations we can recover the first coordinate of **s**, and similarly for other coordinates.
- Algebraic algorithm. This follows from the work of Arora and Ge [AG11]. In this algorithm, we need to bound an interval for the error and deduce a polynomial such that all the error will satisfy this equation. Then after getting enough samples, we can get the secret key **s**.
- Lattice decoding attack. In this algorithm, we first transform LWE into a BDD problem, and use lattice reduction algorithm (such as BKZ [SE94], LLL [LLL$^+$82] to get the original secret key **s**.

In this thesis, we only introduce the naive algorithm and Arora-Ge algorithm.

2.4.1 Naive Algorithm

From now on, we assume that the hypothesis of Theorem 1 is satisfied. It is easy to see that the matrix $\mathbf{A}$ is then of rank n with overwhelming probability (indeed, that probability is exactly $(1-q^{-m})(1-q^{1-m})\cdots(1-q^{n-1-m}) \geq 1-q^{n-m}$, and this can be used to deduce a "naive" algorithm for binary error LWE in time $O^*(2^n)$, essentially by guessing n coefficients of the error vector $\mathbf{e}$.

More precisely, since $\mathbf{A}$ is full rank, one can assume without loss of generality that its first n rows form an invertible square submatrix $\mathbf{A}_0$. An algorithm for binary error LWE is then as follows: guess the vector $\mathbf{e}_0 \in \{0,1\}^n$ consisting of the first n coefficients of $\mathbf{e}$; then deduce the corresponding $\mathbf{s} = \mathbf{A}_0^{-1}(\mathbf{b}_0 - \mathbf{e}_0)$, and check that $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{s}$ is indeed in $\{0,1\}^m$. The check is performed in $poly(n)$ time, and by Theorem 1, there is with overwhelming probability a unique $\mathbf{e}_0 \in \{0,1\}^n$ passing this check, which corresponds to the unique solution $\mathbf{s}$. Trying all possibilities yields an algorithm in $O^*(2^n)$ time.

2.4.2 Arora-Ge algorithm

In a paper published at ICALP 2011, Arora and Ge proposed an algebraic approach to the LWE problem, which essentially amounts to expressing LWE as a system of polynomial equations, and then solving that system by unique linearization techniques. In the case of binary error LWE, the polynomial system is a system of multivariate quadratic equations, which can be solved in polynomial time by linearization when the number m of samples exceeds about $n^2/2$.

More precisely, solving an instance $(\mathbf{A}, \mathbf{b})$ of the binary error LWE problem amounts to finding a vector $\mathbf{s} \in \mathbb{Z}_q^n$ (which we have seen is uniquely determined) such that for $i = 1, \ldots, m$, we have:

$$b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle \in \{0,1\},$$

where the vectors $\mathbf{a}_i$ are the rows of $\mathbf{A}$, and the scalars b_i the coefficients of $\mathbf{b}$. The idea of Arora and Ge is to rewrite that condition as:

$$\big(b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle\big) \cdot \big(b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle - 1\big) = 0,$$

which is a quadratic equation in the coefficients $s_1, \ldots, s_n$ of $\mathbf{s}$.

In general, solving a multivariate quadratic system is hard. However, it becomes easy when many equations are available. Arora and Ge propose to solve this system using a simple linearization technique: replace all the monomials appearing in the system by a new variable.

There are $\binom{n+2}{2} = (n+2)(n+1)/2$ monomials of degree at most 2. Therefore, if the number of samples m is at least $(n+2)(n+1)/2$, linearizing the quadratic system

should yield a full rank linear system with high probability, and the secret $\mathbf{s}$ can be recovered by solving this linear system. This takes time $O\left(\binom{n+2}{2}^{\omega}\right) = O(n^{2\omega})$, and therefore shows that binary error LWE can be solved in polynomial time given $m \approx n^2/2$ samples.

However, many applications of LWE-like problems only give out much fewer than $\Theta(n^2)$ samples. For example, public-key encryption schemes based on LWE-like problems often have a public key consisting of $O(n \log q)$ samples (or in some cases, just $O(n)$ samples). It is therefore interesting to analyze how the complexity of binary error LWE varies as the number of available samples decreases.

2.5 Function Family

Informally speaking, a function family is a probability distribution $\mathcal{F}$ over a set of functions $\mathcal{F} \subset (X \to Y)$ with common domain X and range Y. Let $\mathcal{X}$ be a probability distribution over the domain X of a function family $\mathcal{F}$. We recall the following standard security notions:

One Wayness: $(\mathcal{F}, \mathcal{X})$ is (t, ϵ)-one-way if for all probabilistic algorithms $\mathcal{A}$ running in time at most t,

$$\Pr\left[f \leftarrow \mathcal{F}, x \leftarrow \mathcal{X} : \mathcal{A}(f, f(x)) \in f^{-1}(f(x))\right] \leq \epsilon$$

Uninvertibility: $(\mathcal{F}, \mathcal{X})$ is (t, ϵ)-uninvertible if for all probabilistic algorithms $\mathcal{A}$ running in time at most t,

$$\Pr[f \leftarrow \mathcal{F}, x \leftarrow \mathcal{X} : \mathcal{A}(f, f(x)) = x] \leq \epsilon$$

Second Preimage Resistance: $(\mathcal{F}, \mathcal{X})$ is (t, ϵ)-second preimage resistant if for all probabilistic algorithms $\mathcal{A}$ running in time at most t,

$$\Pr\left[f \leftarrow \mathcal{F}, x \leftarrow \mathcal{X}, x' \leftarrow \mathcal{A}(f, x) : f(x) = f\left(x'\right) \wedge x \neq x'\right] \leq \epsilon$$

Pseudorandomness: $(\mathcal{F}, \mathcal{X})$ is (t, ϵ)-pseudorandom if the distributions $\{f \leftarrow \mathcal{F}, x \leftarrow \mathcal{X} : (f, f(x))\}$ and $\{f \leftarrow \mathcal{F}, y \leftarrow \mathcal{U}(Y) : (f, y)\}$ are (t, ϵ)-indistinguishable.

Lossy Function Families: Lossy functions, introduced in [PW11], are usually defined in the context of trapdoor function families, where the functions are efficiently invertible with the help of some trapdoor information, and therefore injective. Here we have a more general definition of lossy function family that is a general framework used to prove the one-wayness of some function.

Definition 2.5.1 (Lossy Function Families [MP13])**.** Let ($\mathcal{L}$, $\mathcal{F}$) be two probability distributions(with possibly different supports) over the same set of (efficiently computable) functions $\mathcal{F} \subset X \to Y$, and let $\mathcal{X}$ be an efficiently sampleable distribution over the domain X. We say that ($\mathcal{L}$, $\mathcal{F}$, $\mathcal{X}$) is a lossy function family if the following properties are satisfied:

- the distributions $\mathcal{L}$ and $\mathcal{F}$ are indistinguishable.
- ($\mathcal{L}$, $\mathcal{X}$) is uninvertible.
- ($\mathcal{F}$, $\mathcal{X}$) is second preimage resistant.

Theorem 11 ([MP13])**.** *Let $\mathcal{F}$ be a family of functions computable in time t'. If $(\mathcal{F}, \mathcal{X})$ is both (t, ϵ)-uninvertible and $(t + t', \epsilon')$-second preimage resistant, then it is also $(t, \epsilon + \epsilon')$-one-way.*

Proof. Let $\mathcal{A}$ be an algorithm running in time at most t and attacking the one-wayness property of $(\mathcal{F}, \mathcal{X}$. Let $f \leftarrow \mathcal{F}$ and $x \leftarrow \mathcal{X}$ be chosen at random, and compute $y \leftarrow \mathcal{A}(f, f(x))$. We want to bound the probability that $f(x) = f(y)$. We consider two cases:

- If $x = y$, then $\mathcal{A}$ breaks the uninvertibility property of $(\mathcal{F}, \mathcal{X})$.
- If $x \neq y$, then $\mathcal{A}'(f, x) = \mathcal{A}(f, f(x))$ breaks the second preimage property of $(\mathcal{F}, \mathcal{X})$.

By assumption, the probability of these two events are at most ϵ and ϵ' respectively. By the union bound,$\mathcal{A}$ breaks the one-wayness property with advantage at most $\epsilon + \epsilon'$. □

Theorem 12 ([MP13])**.** *Let $\mathcal{F}$ and $\mathcal{F}'$ be any two indistinguishable, efficiently computable function families, and let $\mathcal{X}$ be an efficiently sampleable input distribution. Then if ($\mathcal{F}$, $\mathcal{X}$) is uninvertible(respectively, second-preimage resistant), then ($\mathcal{F}'$, $\mathcal{X}$) is also uninvertible(resp., second preimage resistant). In particular, if ($\mathcal{L}$, $\mathcal{F}$, $\mathcal{X}$) is a lossy function family, then ($\mathcal{L}$, $\mathcal{X}$) and ($\mathcal{F}$, $\mathcal{X}$) are both one-way.*

Proof. Assume that $(\mathcal{F}, \mathcal{X}$ is uninvertible and that there exists an efficient algorithm $\mathcal{A}$ breaking the uninvertibility property of $(\mathcal{F}', \mathcal{X}$. Then $\mathcal{F}$ and $\mathcal{F}'$ can be efficiently distinguished by the following algorithm $D(f)$: choose $x \leftarrow \mathcal{X}$, compute $x' \leftarrow A(f, f(x))$, and accept if $\mathcal{A}$ succeeded, i.e., if $x = x'$.

Next, assume that $(\mathcal{F}, \mathcal{X}$ is second preimage resistant, and that there exists an efficient algorithm $\mathcal{A}$ by breaking the second preimage resistance property of $(\mathcal{F}', \mathcal{X}$. Then $\mathcal{F}$ and $\mathcal{F}'$ can be efficiently distinguished by the following algorithm $D(f)$: choose $x \leftarrow \mathcal{X}$, compute $x' \leftarrow A(f, f(x))$, and accept if $\mathcal{A}$ succeeded, i.e., if $x \neq x'$ and $f(x) = f(x')$.

It follows that if ($\mathcal{L}$, $\mathcal{F}$, $\mathcal{X}$) is a lossy function family,($\mathcal{L}$, $\mathcal{X}$) and ($\mathcal{F}$, $\mathcal{X}$) are both uninvertible and second preimage resistant. Then by theorem 11, they are also one-way. □

Now we introduce two special family of functions, which are the fundamental blocks of lattice cryptography. Both families are parametrized by three integers m, n and q, and a set $X \subset \mathbb{Z}^m$ of short vectors. Usually n serves as a security parameter and m,q are functions of n.

SIS Function Family The Short Integer Solution function family SIS(m, n, q, X) is the set of all functions f_A indexed by $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with domain $X \subseteq \mathbb{Z}^m$ and range $Y = \mathbb{Z}_q^n$ defined as $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{Ax} \bmod q$.

LWE Function Family The Learning With Error function family LWE(m, n, q, X) is the set of all functions g_A indexed by $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with domain $\mathbb{Z}_q^n \times X$ and range $Y = \mathbb{Z}_q^m$, defined as $g_{\mathbf{A}}(\mathbf{s}, \mathbf{x}) = \mathbf{A}^T\mathbf{s} + \mathbf{x} \bmod q$. The reason that we introduce SIS function family here is that in later proof, we first prove the one-wayness of SIS function family, and then use the equivalence of SIS function and LWE function(with respect to some parameter) to show that LWE function family is also one-way. The following theorem says that SIS function and LWE function families are essentially equivalent with respect to some specific parameter setting.

Equivalence of SIS and LWE

Theorem 13 ([Mic10], [MM11]). *For any $n, m \geq n + \omega(logn), q$, and distribution $\mathcal{X}$ over $\mathbb{Z}^m$, the LWE(m, n, q) function family is one-way (resp. pseudorandom, or uninvertible) with respect to input distribution $U(\mathbb{Z}_q^n) \times \mathcal{X}$ if and only if the SIS(m, m - n, q) function family is one-way(resp. pseudorandom , or uninvertible) with respect to the input distribution $\mathcal{X}$.*

Pseudorandomness of SIS Function In order to construct a lossy function family in later proof, we also need the pseudorandomness of SIS function with respect to some specific parameters, which can be derived by the assumption that worst-case SIVP problem is hard.

Theorem 14 ([MP13]). *For any positive m, n, δ, q such that $\omega(\log n) \leq m - n \leq n^{O(1)}$ and $2\sqrt{n} < \delta < q < n^{O(1)}$, if q has no divisors in the range $((\delta/\omega_n)^{1+n/k}$, $\delta \cdot \omega_n)$, then the SIS$(m, m - n, q)$ function family is pseudorandom with respect to input distribution $D_{\mathbb{Z},\delta}^m$, under the assumption that no (quantum) algorithm can efficiently sample(up to negligible statistical errors) $D_{\wedge,\sqrt{2n}q/\delta}$. In particular, assuming the worst-case(quantum) hardness of SIVP$_{n\omega_n q/\delta}$ on n-dimensional lattices, the*

SIS$(m, m-n, q)$ *function family is pseudorandom with respect to input distribution* $D^m_{\mathbb{Z},\delta}$.

2.6 Sample-Time Trade-off for Binary Error LWE

As we already show in previous chapters, binary error LWE can be attacked in polynomial time given $\Theta(n^2)$ samples, and binary error LWE is as hard as worst-case lattice problems with less than $n + O(n/\log n)$ samples. A natual question would be, how about the case when the number of samples is $n^{1+\alpha}$ for $0 < \alpha < 1$. In this chapter, we will use an approach that we call Macaulay matrix method to get a sample-time trade-off for the binary error LWE.

2.6.1 Hilbert's Nullstellensatz for Arora–Ge

Slightly informally, Hilbert's Nullstellensatz essentially states that the ideal generated by a family of polynomials $f_1, \ldots, f_m \in \mathbb{Z}_q[X_1, \ldots, X_n]$ coincides with the ideal of polynomials that vanish on the set $V(f_1, \ldots, f_m)$ of solutions of the polynomial system: Now consider the application of Hilbert's Nullstellensatz to the polynomial system arising from Arora and Ge's approach to binary error LWE. That system is of the form:

$$\begin{cases} f_1(s_1, \ldots, s_n) = 0 \\ \qquad\qquad\qquad \vdots \\ f_m(s_1, \ldots, s_n) = 0 \end{cases}$$

where $f_1, \ldots, f_m \in \mathbb{Z}_q[X_1, \ldots, X_n]$ are known quadratic polynomials. By Theorem 1, the set $V(f_1, \ldots, f_m)$ of solutions of that system is reduced to a single point:

$$V(f_1, \ldots, f_m) = \big\{(s_1, \ldots, s_n)\big\} = \big\{\vec{s}\big\},$$

namely, the unique solution of the binary error LWE problem. It follows[1] that the ideal $I = (f_1, \ldots, f_m) \subset \mathbb{Z}_q[X_1, \ldots, X_n]$ generated by the polynomials f_i coincides with the ideal of polynomial functions vanishing on $\big\{\mathbf{s}\big\}$, which is just $(X_1 - s_1, \ldots, X_n - s_n)$.

[1] We are sweeping two technicalities under the rug. First, the set of solutions considered in the Nullstellensatz should really be computed over the algebraic closure of the base field; however, it is easy to see that the argument of Theorem 1 applies similarly to show uniqueness even for solutions on extensions of $\mathbb{Z}_q$. Second, the Nullstellensatz actually describes the *radical* of the ideal $(f_1, \ldots, f_m)$, but it is clear that this ideal is already radical with overwhelming probability.

Macaulay matrix

Definition 2.6.1. Macaulay matrix Let $f_1, \ldots, f_m \in Z_q[x_1, \ldots, x_n]$ The Macaulay matrix M of degree d is defined as: list "horizontally" all the degree d monomials from smallest to largest sorted by some fixed admissible monomial ordering. The smallest monomial comes last. Multiply each f_i by all monomials $t_{i,j}$ of degree $d - d_i$ where $d_i = deg f_i$. Finally, construct the coefficient matrix.

Theorem 15. *Let* $\mathbf{f} = (f_1, \ldots, f_m) \in (Z_q[x_1, \ldots, x_n])^m$ *and* $<$ *be a monomial ordering. There exists a positive integer D for which Gaussian elimination on all* $M = (f_1, \ldots, f_m)$ *matrices for d,* $1 \leq d \leq D$ *computes Gröbner basis of* $\langle f_1, \ldots, f_m \rangle$ *w.r.t* $<$*. The degree D is called the degree of regularity of* $f_1, \ldots, f_m$*.*

Suppose that we have a system of polynomials equations:

$$\begin{aligned} f_1(s_1, s_2 \cdots s_n) &= 0 \\ f_2(s_1, s_2 \cdots s_n) &= 0 \\ &\cdots \\ f_m(s_1, s_2 \cdots s_n) &= 0 \end{aligned}$$

where $(s_1, s_2 \cdots s_n)$ are the unknown variables that correspond to the components of secret key s. Then we can multiply these equations with any monomials degree less than a particular number D, getting more equations. For instance, if D=1, we can multiply these equations with $s_1, s_2 \cdots s_n$:

$$\begin{aligned} s_1 \times f_1(s_1, s_2 \cdots s_n) &= 0 \\ s_1 \times f_2(s_1, s_2 \cdots s_n) &= 0 \\ &\cdots \\ s_1 \times f_m(s_1, s_2 \cdots s_n) &= 0 \\ &\cdots \\ s_n \times f_1(s_1, s_2 \cdots s_n) &= 0 \\ s_n \times f_2(s_1, s_2 \cdots s_n) &= 0 \\ &\cdots \\ s_n \times f_m(s_1, s_2 \cdots s_n) &= 0 \end{aligned}$$

After getting $\binom{n+D}{n}$equations, in a similar way with Arora-Ge algorithm, we do linearization, make new variables for each monomial and solve the new system of linear equations. The only question left is that how could we guarantee that after linearization, there is a unique solution. In other words, we need to determine D.

Hilbert's Nullstellensatz Given some polynomials $f_1, \ldots, f_m \in \mathcal{K}[x_1, \ldots, x_n]$, The Consistency Question is: Does the system of these polynomial equations, say

$$S = \begin{cases} f_1 = 0 \\ f_2 = 0 \\ \cdots \\ f_m = 0 \end{cases}$$

has a solution in $\mathcal{K}$? HN helps in answering this question. In its weak form, also known as Weak Hilbert's Nullstellensatz (WHN), it gives us a certificate when this system has no solution.

Theorem 16. *Let $f_1, \ldots, f_m \in \mathcal{K}[x_1, \ldots, x_n]$, then the system*

$$S = \begin{cases} f_1 = 0 \\ f_2 = 0 \\ \cdots \\ f_m = 0 \end{cases}$$

will have no solution in $\mathcal{K}$ iff $\exists g_1, g_2, \ldots, g_m \in \mathcal{K}[x_1, \ldots, x_n]$ such that $\sum_{i=1}^{m} f_i g_i = 1$.

Besides, from Hilbert's Nullstellensatz, we can know that if given large enough D, which we mentioned previously, the new system of equations that we construct by multiplication will have one unique solution.

Semi-regularity: It turns out that if assuming semi-regularity, we have a good formula for D. Let $m > n$, and $f_1, \ldots, f_m \in Z_q[x_1, \ldots, x_n]$ be homogeneous polynomials of degree $d_1, \ldots, d_m$ respectively and I the ideal generated by these polynomials. The system is called to be a semi-regular system if the Hilbert series w.r.t the grevlex order $H_I(z) = \left[\frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n}\right]_+$, where $[S]_+$ denotes the series obtained by truncating S before the index of its first non-positive coefficient.

2.6.2 Gröbner basis

Although we don't really use Gröbner basis in our analysis, our method is, in some way, essentially similar with Gröbner basis. Therefore, we also give a short introduction to Gröbner basis here.

Gröbner basis Gröbner basis is a very useful and fundamental tool in commutative algebra to solve a system of non-linear polynomial equations over a finite field. We consider polynomials in $K[x] = K[x_1, x_2....x_n]$.

Definition 2.6.2. A Gröbner basis of an ideal $I \subset K[\mathbf{x}]$ for a given monomial ordering is a finite set $B \subset \mathcal{I}$ such that any $f \in \mathcal{J}$ reduces to 0 by B. The basis is called reduced when the $f_i's$ all have leading coefficient 1 and when none of the $f_i's$ involves a monomial which reduces by $B \setminus \{f_i\}$.

Complexity of computing a Gröbner basis The complexity of computing a Gröbner basis is bounded by the complexity of performing Gaussian elimination on the Macaulay matrix in some degree D. There are several algorithms of computing a Gröbner basis with degree of regularity, such as Buchberger algorithm, F_4, F_5 algorithm [Fau99, BCLA82]. The complexity of computing a Gröbner basis would be: $O\left(\binom{n+d}{d}^{\omega}\right)$, where $2 \leq \omega < 3$ is the linear algebra constant, and d is the degree of semi-regularity of the system.
Generally speaking, it is very difficult to compute the degree of regularity of a polynomial system. But there is a good formula when assuming semi-regularity of the polynomial system.

2.6.3 Arora-Ge attack with Macaulay matrix method on binary error LWE

Recall that solving LWE is actually equivalent to computing a Gröbner basis [ACF+14] for a system of polynomials. Besides, the complexity of computing a Gröbner basis would be: $O\left(\binom{n+d}{d}^{\omega}\right)$, where $2 \leq \omega < 3$ is the linear algebra constant, and d is the degree of semi-regularity of the system. Therefore, in order to estimate the time complexity of binary error LWE attack, we only need to compute d_{reg} for this polynomial system.

Theorem 17. *[BFSY05] For $m = n + k$ ($k > 1$ fixed) quadratic equations in n variables, the degree of regularity d_{reg} behaves asymptotically like*

$$d_{reg} \sim \frac{m}{2} \tag{2.2}$$

The time complexity for binary error LWE would be $O\left(\binom{n+\frac{m}{2}}{\frac{m}{2}}^{\omega}\right)$, which is not in polynomial time.

Theorem 18. *For m = ϵn^2 (ϵ is a constant) quadratic equations in n variables, the degree of regularity d_{reg} behaves asymptotically like*

$$d_{reg} \sim \frac{1}{8\epsilon} \tag{2.3}$$

The time complexity for binary error LWE would be $O\left(\begin{pmatrix} n+\frac{1}{8\epsilon} \\ \frac{1}{8\epsilon} \end{pmatrix}^{\omega}\right)$*, which is in polynomial time.*

Theorem 19. *For m = $n^{1+\alpha}$ (α is a constant between 0 and 1) quadratic equations in n variables, the degree of regularity d_{reg} behaves asymptotically like*

$$d_{reg} \sim \frac{1}{8}n^{1-\alpha} \tag{2.4}$$

The time complexity for binary error LWE would be $O\left(\begin{pmatrix} n+\frac{1}{8}n^{1-\alpha} \\ \frac{1}{8}n^{1-\alpha} \end{pmatrix}^{\omega}\right)$*, which means that when α is smaller, the time complexity grows larger.*

Now we are going to prove these first two theorems. The third theorem is quite similar, which we will not give the proof here.

Proof

Case m = n + k(k is a constant)

Proof. Denote h_d as the d-th coefficient of Hilbert series.

$$\mathrm{H}_{m,n}(z) = \frac{(1-z^2)^m}{(1-z)^n} = \sum_{d=0}^{\infty} h_d z^d \tag{2.5}$$

where the integration path enclose the origin and there are no other singularity of $H_{m,n}(z)$. Take d-th derivative for equation (1) and using Cauchy Integral formula for derivatives, we can get

$$\mathcal{I}_n(d) = \frac{1}{2i\pi}\oint H_{m,n}(z)\frac{dz}{z^{d+1}} = \frac{1}{2i\pi}\oint e^{nf(z)}dz \tag{2.6}$$

Then write the equation in another way

$$\mathcal{I}_n(d) = \frac{1}{2i\pi}\oint \underbrace{(1-z)^{m-n}}_{g(z)}\underbrace{(1+z)^m z^{-d-1}}_{F(z)=e^{nf(z)}}dz \tag{2.7}$$

$$\mathcal{I}_n(d) = \frac{1}{2i\pi} \oint g(z) e^{nf(z)} dz \tag{2.8}$$

Then we get

$$f'(z) = \frac{m}{1+z} - \frac{d+1}{z} \tag{2.9}$$

The saddle point is

$$z_0 = \frac{1}{\frac{m}{d+1} - 1} \tag{2.10}$$

The approximation is

$$\mathcal{I}_n(d) \sim \frac{(1+z_0)^{m+1} (1-z_0)^{m-n}}{\sqrt{2\pi} z_0^{d+1/2} m^{1/2}} \tag{2.11}$$

It vanishes only if $z_0 = 1$, i.e.

$$d_{reg} \sim \frac{m}{2} \tag{2.12}$$

□

Case m = ϵn^2 (ϵ is a constant)

Proof. Denote h_d as the d-th coefficient of Hilbert series.

$$\mathrm{H}_{m,n}(z) = \frac{(1-z^2)^m}{(1-z)^n} = \sum_{d=0}^{\infty} h_d z^d \tag{2.13}$$

where the integration path enclose the origin and there are no other singularity of $H_{m,n}(z)$. Take d-th derivative for equation (1) and using Cauchy Integral formula for derivatives, we can get

$$\mathcal{I}_n(d) = \frac{1}{2i\pi} \oint H_{m,n}(z) \frac{dz}{z^{d+1}} = \frac{1}{2i\pi} \oint e^{nf(z)} dz \tag{2.14}$$

Then determine $f(z)$

$$\mathcal{I}_n(d) = e^{nf(z)} dz = \frac{1}{2i\pi} \oint g(z) e^{nf(z)} dz \tag{2.15}$$

Then we get

$$e^{nf(z)} = \frac{(1-z)^{m+n}(1+z)^m}{z^{d+1}} \tag{2.16}$$

Then we get $f(z)$

$$nf(z) = (m-n)\log(1-z) + m\log(1+z) - (d+1)\log z \tag{2.17}$$

Compute $f'(z)$

$$nf'(z) = \frac{n-m}{1-z} + \frac{m}{1+z} - \frac{d+1}{z} \tag{2.18}$$

Let $f'(z) = 0$

$$(n-2m+d+1)z^2 + nz - (d+1) = 0 \tag{2.19}$$

If Δ of this equation is not zero, it means that there are two distinct saddle points. The contribution of these two saddle points to the integral are conjugate values whose sum does not vanish. Hence the two saddle points must be identical, which means that $\Delta = 0$

$$\Delta = 4(d+1)^2 + 4(n-2m)(d+1) + n^2 = 0 \tag{2.20}$$

Solving this equation, we get

$$d+1 = m - \frac{n}{2} - \sqrt{m(m-n)} \tag{2.21}$$

Substitute m = ϵn^2

$$d+1 = \epsilon n^2 - \frac{n}{2} - \epsilon n^2\sqrt{1 - \frac{1}{\epsilon n}} \tag{2.22}$$

Using taylor expansion

$$d+1 \sim \frac{1}{8\epsilon} \tag{2.23}$$

□

2.7 Hardness of LWE with Non-uniform Binary Error

In this chapter we propose a variant of binary error LWE that we call non-uniform binary error LWE and analyze the hardness of non-uniform binary error LWE. We generalize the uniform case to the nonuniform case, where the error rate is p, thus having the following two results:

- Case 1: We prove that non-uniform binary error LWE is as hard as worst-case lattice problems provided that the number of samples is restricted. This is a generalization of Micciancio and Peikert's hardness proof for uniform binary error LWE.
- Case 2: When the error rate p is a function of n such that $p(n) = 1/n^\alpha$ for any constant $\alpha > 0$, we propose a simple algorithm to give some attacks against non-uniform binary error LWE.

For case 1, we proceed similarly to [MP13], by constructing a lossy function family with respect to the non-uniform input distribution χ. However, since they are dealing with a uniform distribution, we use some computation and estimate to overcome the difficulty of transforming from uniform case to non-uniform case. The basic idea is as follows:

- Construct two indistinguishable function families $\mathcal{F}$ = SIS$(m, m-n, q)$ and $\mathcal{L}$ = SIS$(l, m-n, q) \circ \mathcal{I}(m, l, \mathcal{Y})$, where $\circ$ means the composition of two functions and $\mathcal{I}(m, l, \mathcal{Y})$ is defined in Definition 2.7.1.
- Prove $(\mathcal{L}, \mathcal{X})$ is uninvertible with respect to input distribution $\mathcal{X}$.
- Prove $(\mathcal{F}, \mathcal{X})$ is second-preimage resistant with respect to input distribution $\mathcal{X}$.
- Use the above three properties to show that $(\mathcal{L}, \mathcal{F}, \mathcal{X})$ is a lossy function family.
- By using Theorem 11 to show that $(\mathcal{L}, \mathcal{X})$ and $(\mathcal{F}, \mathcal{X})$ are both one-way, so SIS$(m, m-n, q)$ is one-way with respect to the input distribution $\mathcal{X}$.
- By using Theorem 13 to show that LWE(m, n, q) is one-way with respect to the input distribution $\mathcal{X}$.

In this construction, they first proved the one-wayness of SIS$(m, m-n, q)$, and then use the equivalence of LWE(m, n, q) and SIS$(m, m-n, q)$ to prove LWE(m, n, q) is also one-way. There is some other work(essentially the same) [DMQ13], which directly reduces uniform error LWE to standard LWE without using the notation of SIS. In this thesis, we stick to the SIS notation.

2.7.1 Hardness of Non-uniform Binary Error LWE with Limited Samples

In order to prove $(\mathcal{L}, \mathcal{F}, \mathcal{X})$ is a lossy function family, we will prove:

- $\mathcal{L}$ is uninvertible with respect to $\mathcal{X}$.
- $\mathcal{F}$ is second preimage resistant with respect to $\mathcal{X}$.
- $(\mathcal{L}, \mathcal{F})$ are indistinguishable.

Statistical Uninvertibility

Lemma 20. Let $\mathcal{L}$ be a family of functions on the common domain $\{0,1\}^m$, we define a non-uniform distribution χ over $\{0,1\}^m$ such that each coefficient $x_i (i = 1, \cdots, m)$ is 1 with probability $p (0 < p < 1)$, and set p' = max$(p, 1-p)$. Then $\mathcal{L}$ is ϵ-uninvertible statistically for $\epsilon = \mathbb{E}_{f \leftarrow \mathcal{L}} (p')^m \cdot |f(X)|$, where $|f(X)|$ means the number of elements in the range and $\mathbb{E}$ means taking the expectation over the choice of f.

Proof. Fix any $f \leftarrow \mathcal{L}$ and choose a input x from the distribution χ. Denote $y = f(x)$. The best attack that the adversary can achieve is to choose the most likely element from the preimage, i.e., the element with the highest conditional probability.

$$\begin{aligned}
&Pr[\text{adversary can invert}] \\
&= \sum_x Pr[x] \cdot Pr[\text{adversary can invert given } f(x)] \\
&= \sum_x Pr[x] \cdot Pr[\text{x is the most likely preimage in } f^{-1}(f(x))] \\
&= \sum_{y \in f(X)} \frac{max_{x \in f^{-1}(y)} Pr(x)}{\sum_{x \in f^{-1}(y)} Pr[x]} \cdot \sum_{x \in f^{-1}(y)} Pr[x] \\
&= \sum_{y \in f(X)} max_{x \in f^{-1}(y)} Pr(x)
\end{aligned}$$

All the possbile probability for sampling x from χ is $p^k \cdot (1-p)^{m-k}$ ($k = 0, 1, 2 \cdots m$), we know that the maximum probability is $(max(p, 1-p))^m$. Then let $p' = max(p, 1-p)$, the result follows.

□

Remark. In the above proof, what we are supposed to do is essentially summing up the $|f(X)|$ highest probabilities. From the properties of binomial distribution, it is easy to know that the highest probability is $(p')^m$, the second largest is $(p')^{m-1} \cdot (1-p')$, etc. To be more formal, we need to compute a minimum k such that

$$\sum_{i=0}^{i=k} \binom{m}{i} \geq |f(X)|$$

After getting k, the success probability of the adversary is upperbounded by

$$\sum_{i=0}^{i=k} \binom{m}{i} (1-p')^i (p')^{m-i}$$

The Central Limit Theorem says that the partial sum $\sum_{k=k_1}^{k=k_2} \binom{m}{k} (1-p')^k (p')^{m-k}$ can be well estimated by gaussian approximation for sufficiently large m. However, there is no simple way to integrate the function $e^{-x^2/2}$, so no closed formula for the partial sum exists. In order to deal with this issue, we use a rough estimate to compute the upper bound.

Definition 2.7.1 ([MP13])**.** For any probability distribution $\mathcal{Y}$ over $\mathbb{Z}^l$ and integer $m \geq l$, let $\mathcal{I}(m, l, \mathcal{Y})$ be the probability distribution over linear functions $[I \mid Y]$:

$\mathbb{Z}^m \to \mathbb{Z}^l$ where I is $l \times l$ identity matrix, and $Y \in \mathbb{Z}^{l\times(m-l)}$ is obtained choosing each column of Y independently at random from $\mathcal{Y}$.

Lemma 21. Let χ be a non-uniform distribution over $\{0,1\}^m$ such that each coefficient $x_i (i = 1, \cdots, m)$ is 1 with probability $p (0 < p < 1)$, $\mathcal{Y} = D^l_{\mathbb{Z},\delta}$ be the discrete Gaussian distribution with parameter $\delta > 0$, p' = max$(p, 1-p)$. Then $\mathcal{I}(m, l, \mathcal{Y})$ is ϵ-uninvertible with respect to the non-uniform distribution χ, for $\epsilon = O(\delta m/\sqrt{l})^l \cdot (p')^m + 2^{-\Omega(m)}$.

Proof. In order to use Lemma 20, we only need to bound the size of the range $f(X)$. Recall that $f = [I \mid Y]$ where $Y \leftarrow D^{l\times(m-l)}_{Z,\delta}$. Since the entries of $Y \in \mathbb{R}^{l\times(m-l)}$ are independent mena-zero subgaussians with parameter δ, by a standard bound from the theory of random matrices, the largest singular value $s_1(Y) = max_{0\neq\mathbf{x}\in\mathbb{R}^m}||Yx||/||\mathbf{x}||$ of Y is at most $\delta \cdot O(\sqrt{l} + \sqrt{m-l}) = \delta \cdot O(\sqrt{m})$, except with probability $2^{-\Omega(m)}$. We now bound the l_2 norm of all vectors in the image $f(X)$. Let $\mathbf{u} = (\mathbf{u_1}, \mathbf{u_2}) \in X$, with $u_1 \in \mathbb{Z}^l$ and $\mathbf{u}_2 \in \mathbb{Z}^{m-l}$. Then

$$\begin{aligned}
||f(\mathbf{u})|| &\leq ||\mathbf{u}_1 + Y\mathbf{u}_2|| \\
&\leq ||\mathbf{u}_1|| + ||Y\mathbf{u}_2|| \\
&\leq (\sqrt{l} + s_1(Y)\sqrt{m-l}) \\
&\leq (\sqrt{l} + \delta \cdot O(\sqrt{m})\sqrt{m-l}) \\
&= O(\delta m)
\end{aligned}$$

The number of integer points in the l-dimensional zero-centered ball of radius $R = O(\delta m)$ can be bounded by a simple volume argument, as $|f(X)| \leq (R+\sqrt{l}/2)^n V_l = O(\delta m/\sqrt{l})^l$, where $V_l = \pi^{l/2}/(l/2)!$ is the volume of the l-dimensional unit ball. From Lemma 20, and considering the event that $s_1(Y)$ is not bounded as above, we get that $\mathcal{I}(m, l, \mathcal{Y})$ is ϵ-uninvertible for $\epsilon = O(\delta m/\sqrt{l})^l \cdot (p')^m + 2^{-\Omega(m)}$. □

Second Preimage Resistance

Theorem 22. *Let χ be a non-uniform distribution over $\{0,1\}^m$ such that each coefficient $x_i (i = 1, \cdots, m)$ is 1 with probability $p (0 < p < 1)$. For any integers m, k, any prime q, the function family SIS(m, k, q) is (statistically) ϵ-second preimage resistant with respect to the non-uniform distribution χ for $\epsilon = 2^m/q^k$.*

Proof. Let $\mathbf{x} \leftarrow \chi$ and $A \leftarrow$ SIS(m, k, q) be chosen at random. We want to evaluate the probability that there exists an $\mathbf{x}' \in \{0,1\}^m \backslash \{\mathbf{x}'\}$ such that $A\mathbf{x} = A\mathbf{x}'(\text{mod } q)$, or equivalently, $A(\mathbf{x} - \mathbf{x}') = \mathbf{0}(\text{mod } q)$. Fix two distinct vectors $\mathbf{x}, \mathbf{x}' \in \{0,1\}^m$ and let $\mathbf{z} = \mathbf{x} - \mathbf{x}'$. Then considering taking the random choice of A, since all coordinates of $\mathbf{z}$ are in the range $z_i \in \{-1, 0, 1\}$ and at least one of them is nonzero, the vectors

$A\mathbf{z}(\text{mod } q)$ is distributed uniformly at random in $(\mathbb{Z}_q)^k$, the probability of $A\mathbf{z} = \mathbf{0}$ (mod q) is $1/q^k$. Therefore, by using union bound(over $\mathbf{x}' \in X\backslash\{\mathbf{x}\}$) for any $\mathbf{x}$, the probability that there is a second preimage $\mathbf{x}'$ is at most $(2^m - 1)/q^k < 2^m/q^k$. □

Indistinguishability of $\mathcal{L}$ and $\mathcal{F}$

Lemma 23. Let $\mathcal{F}$ = SIS$(m, m-n, q)$ and $\mathcal{L} = \text{SIS}(l, m-n, q) \circ \mathcal{I}(m, l, \mathcal{Y})$, where $\mathcal{I}(m, l, \mathcal{Y})$ is defined in Definition 2.7.1. If SIS$(l, m-n, q)$ is pseudorandom with respect to the distribution $\mathcal{Y}$, then $\mathcal{L}$ and $\mathcal{F}$ are indistinguishable.

Proof. Choose a random input $\mathbf{x} \in \mathbb{Z}^m$. According to the definition of $\mathcal{F}$ and $\mathcal{L}$

$$\mathcal{L} : \mathbf{x} \to A[I|Y]\mathbf{x} \text{ mod } q$$
$$\mathcal{F} : \mathbf{x} \to [A_1', A_2']\mathbf{x} \text{ mod } q$$

With the property of block matrix multiplication, A can be divided into two blocks: A_1 is a $l \times l$ matrix, A_2 is a $m - n - l \times l$ matrix, so we have

$$\mathcal{L} : \mathbf{x} \to [A_1, A_2Y]\mathbf{x} \text{ mod } q$$
$$\mathcal{F} : \mathbf{x} \to [A_1', A_2']\mathbf{x} \text{ mod } q$$

Since A_1 and A_1' are uniformly random chosen, A_1 and A_1' are indistinguhishable. Recall that SIS$(l, m-n, q)$ is pseudorandom with respect to the distribution $\mathcal{Y}$, thus A_2Y is indistinguishable from A_2'. Then we can conclude that $\mathcal{L}$ and $\mathcal{F}$ are indistinguishable. □

One-wayness

Theorem 24. *Let q be a prime modulus and let χ be a non-uniform distribution over $\{0,1\}^m$ such that each coefficient $x_i (i = 1, \cdots, m)$ is 1 with probability $p (0 < p < 1)$, $p' = max(p, 1-p)$, and $\mathcal{Y}$ be the discrete Gaussian distribution $\mathcal{Y} = D^l_{\mathbb{Z},\delta}$ over $\mathbb{Z}^l$, where $l = m - n + k$ for some $0 < k \leq n \leq m$. If SIS$(l, m-n, q)$ is pseudorandom with respect to the discrete Gaussian distribution $\mathcal{Y} = D^l_{\mathbb{Z},\delta}$, then SIS$(m, m-n, q)$ is $(2\epsilon + 2^{-\Omega(m)})$-one-way with repsect to the input distribution χ if*

$$(C'\delta m/\sqrt{l})^l/\epsilon \leq 1/(p')^m \text{ and } 2^m \leq \epsilon \cdot (q)^{m-n}$$

where C' is universal constant in big O notation in Lemma 21.

Proof. We will prove that $(\mathcal{L}, \mathcal{F}, \mathcal{X})$ is a lossy function family, where $\mathcal{F} = \text{SIS}(m, m-n, q)$ and $\mathcal{L} = \text{SIS}(l, m-n, q) \circ \mathcal{I}(m, l, \mathcal{Y})$. We need to prove the following three things:

- $\mathcal{L}, \mathcal{F}$ are indistinguishable.
- $\mathcal{L}$ is uninvertible with respect to $\mathcal{X}$.
- $\mathcal{F}$ is second preimage resistant with respect to $\mathcal{X}$.

It follows from Lemma 22 that $\mathcal{F}$ is second-preimage resistant with respect to χ. The indistinguishability of $\mathcal{L}$ and $\mathcal{F}$ follows from Lemma 23. By lemma 21, we have the uninvertibility of $\mathcal{L}$. With the three properties of lossy function family, we conclude that $(\mathcal{L}, \mathcal{F}, \mathcal{X})$ is a lossy function family. Then from the property of lossy function family with theorem 12, this theorem is proved.

□

Instantiation for the LWE parameter

Theorem 25 (LWE Parameter)**.** *Let* $0 < k \leq n \leq m$, $0 < p < 1$, $p' = max(p, 1-p)$, $l = m - n + k$, $1/p' \geq (Cm)^{l/m}$ *for a large enough universal constant* C*, and* q *be a prime such that* $max(3\sqrt{k}, 8^{m/(m-n)}) \leq q \leq k^{O(1)}$*. Let* χ *be a non-uniform distribution over* $\{0,1\}^m$ *such that each coefficient* $x_i (i = 1, \cdots, m)$ *is 1 with probability* p*, the LWE*(m, n, q) *function family is one-way with repsect to the distribution* $U_{\mathbb{Z}_q^n} \times \chi$*. In particular, these conditions can be satisfied by setting* $k = n/(c_2 \log_{1/p'} n)$, $m = n(1 + 1/(c_1 \log_{1/p'} n))$*, where* $c_1 > 1$ *is any constant, and* c_2 *such that* $1/c_1 + 1/c_2 < 1$.

Proof. We prove the one-wayness of SIS$(m, m-n, q)$(equivalently, LWE(m, n, q) because of theorem 13) using theorem 24. Thus we need to satisfy the two requirements:

$$(C'\delta m/\sqrt{l})^l/\epsilon \leq 1/(p')^m \text{ and } 2^m \leq \epsilon \cdot (q)^{m-n}$$

Set $\delta = 3\sqrt{k}$ and since $l \geq k$ and the primality of q, the first requirement can be simplified to $\frac{(3C'm)^l}{(1/p')^m} < \epsilon$. Since we have $1/p' \geq (Cm)^{l/m}$, so $(1/p')^m \geq (Cm)^l$. Let $C = 4C'$, we get that $\frac{(3C'm)^l}{(1/p)^m} \leq (3/4)^{-l} \leq (3/4)^{-k}$ is exponentially small in k, so the first inequality is satisfied. Since $q > 8^{m/(m-n)}$, the second inequality is also satisfied.

Besides, we also need to prove the pseudorandomness of SIS$(l, m-n, q)$ with respect to discrete Gaussian distribution $\mathcal{Y} = D^l_{\mathbb{Z},\delta}$, which can be based on the hardness of SIVP on k-dimensional lattice using Theorem 14. After properly renaming the variables, and using $\delta = 3\sqrt{k}$, the requirement becomes $\omega(\log k) \leq m - n \leq k^{O(1)}, 3\sqrt{k} < q < k^{O(1)}$. The corresponding assumption is the worst-case hardness of SIVP$_\gamma$ on k-dimensional lattices, for $\gamma = \tilde{O}(\sqrt{k}q)$. For the particular instantiation, let $m = n(1 + 1/(c_1 \log_{\frac{1}{p'}} n))(c_1 > 1)$, $k = n/(c_2 \log_{\frac{1}{p'}} n)$(c_2 is a positive

constant such that $1/c_1 + 1/c_2 < 1$). The requirement $1/p' \geq (Cm)^{l/m}$ is equivalent to $m \geq l \log_{1/p'} Cm$. Since we can do a asymptotic analysis

$$\begin{aligned} l &= m - n + k \\ &= (1/c_1 + 1/c_2)n / \log_{1/p'} n \end{aligned}$$

And

$$\begin{aligned} &\log_{1/p'} Cm = \log_{1/p'} Cn(1 + 1/log_{1/p'} n) \\ &\approx \log_{1/p'} n + \log_{1/p'} C \end{aligned}$$

So we have

$$l \log_{1/p'} Cm \approx (1/c_1 + 1/c_2)n(1 + \log_{1/p'} C / \log_{1/p'} n)$$

When $(1/c_1 + 1/c_2) < 1$, $m \geq l \log_{1/p'} Cm$ asymptotically. This concludes the proof. □

2.7.2 Attacks Against Non-uniform Binary Error LWE

In the previous section, we show that when the number of samples is strongly restricted, non-uniform binary error LWE is as hard as worst-case lattice problems. However, if relaxing the number of samples, this is not the case. In this section, we consider the case where the error rate is a function of n such that $p = 1/n^{\alpha} (\alpha > 0)$. We show an attack against LWE with non-uniform binary error given $O(n)$ samples. Note that this doesn't contradict with the previous result, because in this big O notation, we have a universal constant C. For instance, if $C = 3$, it means that we can get some attack with $3n$ samples, but have security guarantee with $n + O(n/\log n)$ samples. The idea behind our attack is quite simple,

- Step 1: Get n samples from the LWE oracle.
- Step 2: By assuming the n samples are all error free, solve the linear equation system.
- Step 3: If failed, go back to step1.

For instance, when the error rate $p = 1/n$, the probability that all samples are error free is:

$$\lim_{n \to \infty} (1 - 1/n)^n = 1/e$$

This means that our algorithm is expected to stop after polynomial times of trials. However, the number of total samples used is not bounded. Therefore, we slightly modified the algorithm as follows:

- Step 1: Get $3n$ samples from the LWE oracle.
- Step 2: Choose $2n$ samples randomly from the 3n samples got in step1.
- Step 3: By assuming the $2n$ samples are all error free, solve the linear equation system.
- Step 4: If failed, go back to step2.

We analyze the following two cases respectively:

- $p = 1/n^{\alpha}$ for any consant $\alpha \geq 1$.
- $p = 1/n^{\alpha}$ for any consant $0 < \alpha < 1$.

and have the following results:

Theorem 26. *By applying the above algorithm, for any positive constant $\alpha \geq 1$, non-uniform binary error LWE with error rate $p = 1/n^{\alpha}$ can be attacked in polynomial time with $O(n)$ samples, and for any positive constant $0 < \alpha < 1$, non-uniform binary error LWE with error rate $p = 1/n^{\alpha}$ can be attacked in subexponential time with $O(n)$ samples.*

Proof. Suppose that there are m errors within the $3n$ samples. The probability that

$2n$ samples are all error free is

$$\begin{aligned}
Pr(\text{success}) &= \frac{\binom{3n-m}{2n}}{\binom{3n}{2n}} \\
&= \frac{(3n-m)!}{(n-m)!(2n)!} \cdot \frac{(2n)!(n!)}{(3n)!} \\
&= \frac{(3n-m)!}{(n-m)!} \cdot \frac{(n!)}{(3n)!} \\
&= \frac{n \cdots (n-m+1)}{3n \cdots (3n-m+1)} \\
&= \prod_{i=0}^{m-1} \frac{n-i}{3n-i} \\
&= (\frac{1}{3})^m \prod_{i=0}^{m-1} \frac{3n-3i}{3n-i} \\
&= (\frac{1}{3})^m \prod_{i=0}^{m-1} (1 - \frac{2i}{3n-i}) \\
&\geq (\frac{1}{3})^m (1 - \frac{2m-2}{3n-m+1})^m \\
&= (\frac{1}{3})^m \text{exp}(-\frac{m(2m-2)}{3n-m+1})
\end{aligned}$$

With tail bound for binomial distribution,

$$Pr(m \geq k) \leq \text{exp}(-nD(\frac{k}{n}||p))$$

where $D(a||p)$ is the relative entropy between an a-coin and a p-coin.

$$D(a||p) = a \log \frac{a}{p} + (1-a) \log \frac{1-a}{1-p}$$

We analyze the result from two perspectives.

- $\alpha \geq 1$.
- $0 < \alpha < 1$.

Case 1: $\alpha \geq 1$ For this case, we set $k = \log n$.

$$\begin{aligned} D(\frac{k}{n}||p) &= D(\frac{\log n}{n}||\frac{1}{n^\alpha}) \\ &= \frac{\log n}{n} \cdot \log(n^{\alpha-1}\log n) + (1 - \frac{\log n}{n}) \log \frac{1 - \frac{\log n}{n}}{1 - \frac{1}{n^\alpha}} \\ &\approx (\alpha - 1)\frac{(\log n)^2}{n} + \frac{\log n}{n} \log\log n + (1 - \frac{\log n}{n})(-\frac{\log n}{n} + \frac{1}{n^\alpha}) \end{aligned}$$

Since $(\alpha - 1)\frac{(\log n)^2}{n}$ is the dominant term,

$$Pr(m \geq \log n) \leq \exp(-nD(\frac{k}{n}||p))$$

is negligible. Thus the probability that 2n samples are all error free is

$$Pr(\text{success}) \geq 1/\text{poly}(n)$$

This means that after repeating step2 and step3 polynomial many times, we can recover the secret key with overwhelming probability.

Case 2: $0 < \alpha < 1$ For this case, we set $k = n^{1-\alpha}\log n$

$$\begin{aligned} D(\frac{k}{n}||p) &= D(\frac{n^{1-\alpha}\log n}{n}||\frac{1}{n^\alpha}) \\ &= D(\frac{\log n}{n^\alpha}||\frac{1}{n^\alpha}) \\ &= \frac{\log n}{n^\alpha}\log\log n + (1 - \frac{\log n}{n^\alpha}) \log \frac{1 - \frac{\log n}{n^\alpha}}{1 - \frac{1}{n^\alpha}} \\ &\approx \frac{\log n}{n^\alpha}\log\log n + (1 - \frac{\log n}{n^\alpha})(-\frac{\log n}{n^\alpha} + \frac{1}{n^\alpha}) \end{aligned}$$

The dominant term is $\frac{\log n}{n^\alpha}\log\log n$, so

$$\begin{aligned} Pr(m \geq n^{1-\alpha}\log n) &\leq \exp(-nD(\frac{k}{n}||p)) \\ &\leq \exp(-n^{1-\alpha}\log n \log\log n) \end{aligned}$$

This probability is negligible, thus we have

$$Pr(\text{success}) \geq 1/\exp(n^{1-\alpha})$$

This means that after repeating step2 and step3 subexponential times, we can recover the secret key with overwhelming probability.

□

Chapter 3

Improving Lattice Attacks on (EC)DSA

This chapter is based on joint work with Thomas Espitau and my supervisors.

3.1 Introduction

A lattice is a discrete group of points in space, which can be defined as the set of all integer linear combinations of a certain set of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ known as a basis. A lattice has infinitely many bases, but so-called "reduced" bases, that consist of short and close to orthogonal vectors, are much more interesting. Lattice reduction, the mathematical problem of finding such bases, has a long history which can be traced back to the 18th century, but gained particular prominence after Lenstra, Lenstra and Lovász [LLL+82] introduced a polynomial-time approximate algorithm for it in 1982 that became known as LLL. Since the advent of LLL, lattice reduction proved to be a powerful tool for cryptanalysis: early examples include attacks on knapsack-based cryptosystems [Sha82] and Coppersmith's small root finding algorithm [Cop97] that broke many variants of RSA in particular.

This work focuses on another major cryptanalytic application of lattice reduction: lattice attacks against (EC)DSA (and related signature schemes like Schnorr's) when bits of the nonce are known. DSA and ECDSA are well-established standards for digital signature based on the discrete logarithm problem, and that involve the use, for each generated signature, of some fresh random value called the *nonce*. It is well-known that if the same nonce is used twice, the adversary can directly compute the private key due to a linear relation between the nonce and the private signing key. Even worse, *partial* information about the nonces of multiple signatures can lead to recovery of the full private key. The original approach to do so, due to Bleichenbacher, actually relied on discrete Fourier analysis techniques [Ble00, DHMP13,

AFG+14b, TTA18, ANT+20], but lattice reduction was also discovered to provide an attack technique, in connection to Boneh and Venkatesan's hidden number problem (HNP) [BV96].

HNP is a number theoretic problem that was originally introduced to establish bit security results for the Diffie–Hellman key exchange. Boneh and Venkatesan showed that it could be seen as a bounded distance decoding (BDD) instance in a lattice, which could be solved with Babai's nearest plane algorithm [Bab86] for suitable parameters. Subsequently, Howgrave-Graham and Smart [HGS01], and later Shparlinski and Nguyen [NS02], observed that the problem of attacking (EC)DSA if some top or bottom bits of the nonces are known is an instance of HNP, and could be attacked using the same lattice techniques. However, when nonce leakage is very small, the attack becomes much more difficult mainly because the hidden lattice vector in BDD is not very close to the target vector. It took significant development in lattice reduction algorithms to advance the state of the art. In 2013, Liu and Nguyen [LN13] were able to attack 160-bit DSA with 2-bit nonce leakage using the BKZ 2.0 algorithm introduced just a few years earlier [CN11], relying on a very high block size of 90, with pruned enumeration as the SVP oracle. In a very recent work [AH21a], Albrecht and Heninger utilize the state-of-the-art lattice reduction algorithm G6K [ADH+19] together with the novel idea of predicate sieving to break new records.

3.1.1 Our Contributions

Lattice attacks on (EC)DSA are in general *all-or-nothing*, in the sense that the attack reveals the entire secret key when it succeeds, and nothing at all otherwise. In contrast, Bleichenbacher's statistical attack, for example, only reveals some bits of the secret key in a single execution; however, it has been observed in previous work that the knowledge of those bits makes subsequent applications of the attack much more efficient.

How the knowledge of some bits of the secret key affects lattice attacks on (EC)DSA, however, does not appear to have been considered in previous work[1]. Perhaps interestingly, we observe that knowledge of some bits of the secret *does* in fact make the attack easier. This results in a simple idea to improve those attacks, which forms the main contribution of this work: *guess* some bits of the secret key, and solve the resulting, easier lattice problem for each possible guess (in other words: carry out an

[1] How the knowledge of certain types of side information on the secret affects the hardness of *lattice problems* like LWE has been considered, e.g., in [DDGR20]. The context of HNP/nonce leakage, however, is very different: for example, the key in our setting is an element of $\mathbb{Z}_q$, as opposed to a vector in LWE; the nature of the hints (bits vs. linear relations) is different; the lattice is very structured (knapsack-like) for HNP, as opposed to random q-ary for LWE; the BDD parameters are totally dissimilar; the analysis in their case depends on Gaussian noise, etc. So the two questions appear to be mostly unrelated.

exhaustive search on those bits).

An interesting feature of this approach is that this reduces the attack to solving many BDD instances with varying target vectors in the *same* lattice, making it possible to rely on various batch-CVP or CVP-with-preprocessing techniques to solve them. At the simplest level, even just carrying out an initial lattice reduction on the original BDD lattice, and then solving all the BDD instances by reduction to SVP using Kannan's embedding technique turns out to be far more efficient than naively solving the SVP instances without the initial common lattice reduction.

Additionally, this approach parallelizes very easily, and has the convenient property of being very easy to simulate (in the sense that one can certainly make the "correct" guess for self-generated instances), which makes its cost easy to predict even for parameters that are impractical to fully run in a short time.

As additional contributions, we also show that the same idea can be applied to guessing additional bits of some of the signature nonces (on top of those already known; this results in a similar, but usually slightly worse, success-time trade-off than guessing bits of the secret key), as well as filtering some of the signatures to construct lattices that are easier to attack (resulting in a data-time trade-off reminiscent to what can be achieved in Bleichenbacher's attack). Furthermore, we carry out experiments on the TPM–FAIL dataset [MSEH20] and apply our techniques to key recovery. While the original attack requires about 40000 signatures, with the method of guessing bits, we are able to recover the secret key with only around 800 signatures, which is comparable to the results achieved in Minerva [JSSS20].

3.1.2 Related Work

The main question that we consider, namely how small of a nonce leakage do we need to recover the signing key in (EC)DSA, has been considered in previous work both for lattice attacks and for Bleichebacher's attack. In the case of lattice attacks, the record-holding works are due to Liu and Nguyen [LN13] and very recently Albrecht and Heninger [AH21a]. In the case of Bleichenbacher's attack, the state of the art is presented in [ANT+20]. We briefly describe below how our results compares to theirs.

Comparison with [LN13] and [AH21a]. Table 3.1 presents typical parameters (in terms of group size and number of known nonce bits) for (EC)DSA, and indicates whether they can be tackled easily with lattice attacks ("Easy"), are considered hard so far with lattices ("Hard"), or have been solved in specific papers. In [LN13] and [AH21a], strong lattice reduction algorithms (BKZ 2.0 and G6K with predicate respectively) are used to attack the "borderline" cases, namely 160-bit modulus with 2-bit nonce leakage, 256-bit modulus with 3-bit nonce leakage and 384-bit modulus

Table 3.1: Tractable parameters for lattice attacks on (EC)DSA.

	Nonce leakage			
Modulus	4-bit	3-bit	2-bit	1-bit
160-bit	Easy	Easy	[LN13], [AH21a], Ours	Hard
256-bit	Easy	[AH21a], Ours	Hard	Hard
384-bit	[AH21a], Ours	Hard	Hard	Hard

with 4-bit nonce leakage. Our approach all makes those borderline cases tractable, but relies on very different techniques and arguably present various advantages, particularly the following:

- whereas [LN13] and [AH21a] are based on specific improvements and modifications of the underlying lattice reduction algorithms, our approach works with any lattice reduction algorithm. In our experiments, we use fplll's implementation of BKZ–30, but any algorithm would work. In particular, it is straightforward to combine our idea with the techniques of those two papers if so desired;

- tailoring the parameters of [LN13] and [AH21a] to a specific problem instance or to the specific computational resources of the attack can be quite challenging; in contrast, due to its straightforward simulatability mentioned above, our approach makes this easy, and makes it possible to quantify the cost of attacking a given problem instance in very concrete terms in advance.

Comparison with Bleichenbacher's attack. Although we come up with an approach to improve lattice attacks with more signatures and in some sense bridge the gap between lattice attacks and Bleichenbacher's, it still requires too many signatures compared with Bleichenbacher's attack. For instance, for 160-bit (EC)DSA with 2-bit nonce leakage, our method requires 2^{27} signatures, while the Bleichenbacher attack requires about 2^{12} signatures for 2-bit leakage case and 2^{27} signatures for the one-bit leakage case [ANT+20]. Besides, with this approach, we still could not attack harder cases, such as 160-bit modulus with 1-bit nonce leakage and 256-bit modulus with 2-bit nonce leakage, which are already tractable using Bleichenbacher's attack [AFG+14b, TTA18, ANT+20]. However, the fact that there exists a way of improving lattice attacks with more signatures might give some ideas for future work. It is still possible that better ways of utilizing more signatures for lattice attacks exist, and we hope that lattice attacks on (EC)DSA could be further improved.

3.2 Preliminaries

3.2.1 Lattices

A *lattice*[2] is an additive subgroup of $\mathbb{Z}^n$ for some $n \geq 0$. For any family of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$ of $\mathbb{Z}^n$, the set:

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \Big\{ \sum_{i=1}^{n} c_i \mathbf{b}_i : c_i \in \mathbb{Z} \Big\}$$

is a lattice, and conversely, any lattice $\mathcal{L} \subset \mathbb{Z}^n$ can be put in that form for some vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m$. In that case, the family $(\mathbf{b}_1, \ldots, \mathbf{b}_m)$ is called a basis of $\mathcal{L}$. We can then represent the lattice $\mathcal{L}$ by the $m \times n$ matrix B whose rows are formed by the vectors $\mathbf{b}_i$, and write $\mathcal{L} = \mathcal{L}(B)$.

A given lattice $\mathcal{L}$ can have infinitely many distinct bases, but they all have the same cardinality m, called the *rank* of $\mathcal{L}$. In this work, we will only consider *full-rank* lattices, whose rank m is equal to n, the dimension of the ambient space. For a full-rank lattice $\mathcal{L}$ with basis matrix B, we define the volume of $\mathcal{L}$ as the quantity:

$$\text{vol}(\mathcal{L}) = |\det(B)|,$$

which does *not* depend on the choice of B.

The Euclidean norm of the shortest non-zero vector in $\mathcal{L}$ is called the first minimum of $\mathcal{L}$ and denoted as $\lambda_1(\mathcal{L})$. More generally, for $1 \leq i \leq n$, the i-th minimum $\lambda_i(\mathcal{L})$ of $\mathcal{L}$ is defined as the minimum radius r such that a ball centered at origin with radius r contains i linearly independent vectors.

It is proved in [Ajt06] that a random n-dimensional lattice satisfies, with high probability,

$$\forall 1 \leq i \leq n, \quad \lambda_i(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} \text{vol}(\mathcal{L})^{1/n}.$$

The approximation factor of a lattice basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is defined as $\frac{\|\mathbf{b}_1\|}{\lambda_1(\mathcal{L})}$ (where $\|\cdot\|$ henceforth denotes the Euclidean norm), and the root Hermite factor is defined as $\left(\frac{\|\mathbf{b}_1\|}{\text{vol}(\mathcal{L})^{1/n}}\right)^{1/n}$.

There are many computational problems related to lattices. The most famous one is the Shortest Vector Problem (SVP for short): given a lattice $\mathcal{L}$, find the shortest vector $\mathbf{v} \in L$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$. Another problem is the Closest Vector Problem (CVP for short): given a lattice $\mathcal{L}$ and a target vector $\mathbf{t}$, find the vector $\mathbf{v} \in L$ such that $\|\mathbf{v} - \mathbf{t}\|$ is minimal.

[2] This is more properly the definition of an *integral* lattice, but integral lattices are the only ones we consider in this work.

There exist efficient lattice algorithms for solving approximate versions of SVP and CVP. For approximate SVP, lattice reduction algorithms such as LLL [LLL+82] and BKZ [SE94] output lattice basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ such that the approximation factor and the root Hermite factor are relatively small. As a result, the first vector $\mathbf{b}_1$ of the reduced basis is a good approximation of the shortest non-zero vector. For approximate CVP, Babai's nearest plane algorithm [Bab86] and variants of it such as [Kle00, GPV08, EK20] can be used to find a relatively close vector when applied after a lattice reduction algorithm.

3.2.2 Hidden Number Problem

The Hidden Number Problem can be described as follows: q, l are fixed integers known to the public and α is a unknown integer in $\mathbb{Z}_q$. For many known random $t \in \mathbb{Z}_q$, we have an oracle $\mathcal{O}_\alpha(t)$ that on input t, outputs (t, u) such that $|\alpha \cdot t - u|_q < q/2^l$, where $|z|_q$ represents the unique integer $0 \leq x < q$ such that $x \equiv z \bmod q$. The goal is to recover the hidden secret key α. Suppose that we have queried the oracle d times and have d pairs (t_i, u_i) $(i = 1, 2, \ldots, d)$, we could transform this into a lattice problem. Construct a lattice $\mathcal{L}$ spanned by the following matrix B:

$$B = \begin{pmatrix} 2^l q & 0 & \cdots & 0 & 0 \\ 0 & 2^l q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^l q & 0 \\ 2^l t_1 & 2^l t_2 & \cdots & 2^l t_d & 1 \end{pmatrix}$$

Since $|\alpha t_i - u_i|_q < q/2^l$, there exists some integer c_i such that $|\alpha t_i - u_i + c_i q| < q/2^l$, so $|2^l \alpha t_i - 2^l u_i + 2^l c_i q| < q$, and $\mathbf{h} = (2^l \alpha t_1 + c_1 2^l q, 2^l \alpha t_2 + c_2 2^l q, \ldots, 2^l \alpha t_d + c_d 2^l q, \alpha)$ is a lattice vector (which we call the hidden lattice vector) in $\mathcal{L}$, and set the target vector $\mathbf{v} = (2^l u_1, 2^l u_2, \ldots, 2^l u_d, 0)$. Denote the difference vector $\mathbf{h} - \mathbf{v}$ as $\mathbf{e}$. Since $|2^l \alpha t_i - 2^l u_i + 2^l c_i q| < q$ $(i = 1, 2, \ldots, d)$, it is easy to know that the absolute value of each coefficient of $\mathbf{e}$ is less than q. Therefore, the Euclidean norm of $\mathbf{e}$ is at most $q\sqrt{d+1}$. When l is not too small, the target vector $\mathbf{v}$ is a close vector to the lattice $\mathcal{L}$, so this becomes a CVP instance (or more precisely, BDD instance). Generally, there are two ways to solve the HNP, i.e., the CVP approaches and SVP approaches. In the original paper by Boneh and Venkatesan, they use the LLL algorithm to reduce the lattice basis and Babai's nearest plane algorithm to find the hidden lattice vector. The LLL reduction can be replaced with BKZ. We can also use CVP enumeration instead of nearest plane algorithm. Besides, another technique, known as Kannan's embedding method [Kan87], transforms the CVP instance into a SVP instance by

embedding the target vector into the original lattice, thus constructing a larger lattice:

$$C = \begin{pmatrix} B & 0 \\ \mathbf{v} & q \end{pmatrix}.$$

Then, we can solve SVP by lattice reduction. In our context, we mainly use Kannan's embedding method to solve HNP.

In practice, there are two subtle technical points that we should take care of:

- We might find $q - \alpha$ instead of the secret key α, since $q - \alpha$ is also a good candidate (this can be easily checked). Therefore, we should check both. Note that the checking time is almost negligible compared with the time in lattice reduction, because the only operation is one scalar multiplication (for ECDSA) and checking consistency with public key.

- Typically in practical attacks, the vector that we want is not the first vector of the reduced basis, so we should check every row of the reduced basis. In other words, the attacks are considered successful if we find the vector in any row of the reduced basis (this is typical in the literature).

3.2.3 (EC)DSA Signature Scheme

Here we only discuss DSA and skip ECDSA, since for the construction of HNP instances, this makes no difference. DSA is an El Gamal-like signature scheme, which is included in Digital Signature Standard (DSS) issued by NIST. DSA can be described as follows.

Parameters. The parameters are p, q, g, where p and q are primes satisfying $q|(p-1)$, $g \in \mathbb{Z}_p^*$ has order q. Besides, we have a hash function h that maps any arbitrary-length string into $\mathbb{Z}_q$. The signing key α is a uniformly random number in $\mathbb{Z}_q^*$ and the public key is $y = g^\alpha \bmod p$.

Signing Phase. To sign a message m, the nonce k is chosen uniformly at random from $\mathbb{Z}_q^*$, and we compute $r = (g^k \bmod p) \bmod q$, and $s = k^{-1}(h(m) + \alpha r) \bmod q$. The signature is the pair (r, s).

Verification Phase. Given a signature pair (r, s) of the message m, if $r = (g^{h(m)s^{-1}} y^{rs^{-1}} \bmod p) \bmod q$, the signature is regarded as valid, otherwise invalid.

3.2.4 Lattice Attacks on (EC)DSA

From the signing phase of (EC)DSA, we already know that $s \equiv k^{-1}(h(m) + \alpha r) \bmod q$, so

$$\alpha r \equiv sk - h(m) \pmod{q}.$$

Now in our case, we have l-bit leakage, which means that we know l LSBs of k. In the case of timing attack, MSB is used, where the construction is very similar but slightly subtle. The difference is that when k has some leading zeroes, $k < q/2^l$ might not be true depending on the order q. For more discussion, see Section 4.3 of [JSSS20]. Denote the value of l LSBs as k_1, then we have $k = 2^l k_2 + k_1$ for some integer $0 \leq k_2 \leq q/2^l$, so:

$$\begin{aligned} \alpha r &\equiv s(2^l k_2 + k_1) - h(m) \pmod{q} \\ \alpha(rs^{-1} - k_1)2^{-l} &\equiv k_2 - 2^{-l}s^{-1}h(m) \pmod{q}. \end{aligned}$$

For simplicity of formulas, we set $k_1 = 0$ (without loss of generality, because we know the value of k_1) and have:

$$\begin{aligned} t &\equiv 2^{-l}s^{-1}r \pmod{q} \\ u &\equiv -2^{-l}s^{-1}h(m) \pmod{q} \\ k_2 &\equiv \alpha t - u \pmod{q}. \end{aligned}$$

Note that both t and u can be computed from all the public available information. Since $0 \leq k_2 < q/2^l$,

$$|\alpha t - u|_q < q/2^l.$$

In this way, we have constructed an HNP instance for (EC)DSA. Then we solve the HNP either by nearest plane algorithm or Kannan's embedding method.

3.2.5 Recentering Technique

In order to further improve the lattice attack on (EC)DSA, there is a well-known technique in the community called recentering [NT12]. It works as follows: since

$$|\alpha t - u|_q < q/2^l,$$

there exists some integer c such that

$$\begin{aligned} 0 \leq \alpha t - u + cq &< q/2^l, \\ -q/2^{l+1} \leq \alpha t - u - q/2^{l+1} + cq &< q/2^{l+1}. \end{aligned}$$

Therefore,

$$|\alpha t - u - q/2^{l+1}|_q < q/2^{l+1}.$$

Now set

$$v = 2^{l+1}u + q.$$

Then we have

$$|\alpha t - v/2^{l+1}|_q < q/2^{l+1}.$$

Suppose that now we have d signatures (r_i, s_i) $(i = 1, \ldots, d)$ and compute the pairs (t_i, u_i) as previously defined. Then construct a lattice $\mathcal{L}$ spanned by the following matrix B:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 1 \end{pmatrix}$$

and everything goes the same.

3.2.6 Projected Lattice

Typically, in standard lattice attacks, we almost always locate the secret key in the second row (which we hope to be the first) of the reduced basis. In order to deal with this issue, [AH21a] makes a modification to the original lattice. Recall that the matrix that we construct is:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 1 \end{pmatrix}$$

With some simple linear combinations of the rows, we could know that $(0, 0, \ldots, 0, q)$ belongs to this lattice. The expected Euclidean norm of the difference vector $\mathbf{e}$ is roughly $\sqrt{\frac{d+1}{3}}q$. With typical parameters such as $d = 85$, $l = 2$, $\|\mathbf{e}\|$ is much larger than q. This means that the difference vector $\mathbf{e}$ will never be the shortest vector in practice. In fact, we can project this lattice orthogonal to $(0, \ldots, 0, q)$ and construct a new lattice:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1(t_d)^{-1} & 2^{l+1}t_2(t_d)^{-1} & \cdots & 2^{l+1}t_{d-1}(t_d)^{-1} & 2^{l+1} \end{pmatrix}.$$

In this new lattice, the hidden vector will be $(|\alpha t_d|_q \cdot 2^{l+1} t_1 (t_d)^{-1} + c_1 2^{l+1} q, \ldots, |\alpha t_d|_q \cdot 2^{l+1} t_{d-1} (t_d)^{-1} + c_d 2^{l+1} q, 2^{l+1} |\alpha t_d|_q)$. The important thing is that the vector $(0, 0, \ldots, 0, q)$ does not belong to the new lattice, so we are able to locate the private key in the first row of the reduced basis.

3.3 Analysis: Modeling Lattice Attacks on (EC)DSA

As previously mentioned, there are "borderline" cases that were considered difficult for standard lattice attacks on (EC)DSA, e.g., 160-bit modulus with 2-bit nonce leakage, 256-bit modulus with 3-bit nonce leakage, 384-bit modulus with 4-bit nonce leakage. One important question about this is: *How difficult are those "borderline" cases?* In this section, we explain this question, quantify the difficulty and give intuitive ideas for our attacks in later sections.

3.3.1 Difficulty When Nonce Leakage is Small

For each HNP inequality, there exists some integer c_i such that

$$|\alpha 2^{l+1} t_i - v_i + c_i 2^{l+1} q| < q.$$

Let the target vector $\mathbf{v} = (v_1, \ldots, v_d, 0)$ and the hidden lattice vector $\mathbf{h} = (\alpha 2^{l+1} t_1 + c_1 2^{l+1} q, \ldots, \alpha 2^{l+1} t_d + c_d 2^{l+1} q, \alpha)$, thus the Euclidean norm of the difference vector $\mathbf{e}$ is upper bounded by $q\sqrt{d+1}$. The volume of this lattice $\mathcal{L}$ is $q^d 2^{(l+1)d}$, and according to Gaussian Heuristic, the Euclidean norm of the shortest vector is roughly

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{d+1}{2\pi e}} (\text{vol})^{\frac{1}{d+1}} \approx \sqrt{\frac{d+1}{2\pi e}} 2^{\frac{(l+1)d}{d+1}} q^{\frac{d}{d+1}}.$$

Therefore, the requirement is that the distance is much smaller than $\lambda_1(\mathcal{L})$:

$$q\sqrt{d+1} < \sqrt{\frac{d+1}{2\pi e}} 2^{\frac{(l+1)d}{d+1}} q^{\frac{d}{d+1}}.$$

After solving this inequality, we get

$$d \geq \frac{\log_2(q)}{l - \log_2(\sqrt{\pi e/2})}.$$

This can be used to estimate the number of signatures needed for the attack to succeed. Table 3.2 is the typical number of signatures needed (just information-theoretically, the attack might not be successful at all) to perform the lattice attack on 160-bit

Table 3.2: Typical number of signatures for 160-bit modulus.

Nonce leakage l	4-bit	3-bit	2-bit	1-bit
Number of signatures d	50	80	100	200

(EC)DSA. Now, we give an intuitive explanation of why lattice attacks against (EC)DSA with small nonce leakage are difficult.

When $l = 3$ and $d = 80$ (this case is regarded as "easy" for lattice attacks), the lattice basis matrix B is:

$$B = \begin{pmatrix} 16q & 0 & \cdots & 0 & 0 \\ 0 & 16q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 16q & 0 \\ 16t_1 & 16t_2 & \cdots & 16t_d & 1 \end{pmatrix}.$$

The Euclidean norm of the first vector is $16q$, and $\|\mathbf{e}\|$ is upper bounded by $q\sqrt{d+1} = 9q$. Therefore, any linear combination of the first d rows will have significantly larger Euclidean norm than $\|\mathbf{e}\|$.

When $l = 2$ and $d = 100$ (this case is regarded as "hard" for standard lattice attacks, but have been solved in specific papers), the Euclidean norm of the first vector is $8q$, and $\|\mathbf{e}\|$ is upper bounded by $q\sqrt{d+1} \approx 10q$. To be a bit more precise, we can compute the expected norm, which is roughly $\sqrt{\frac{100}{3}q^2} \approx 6q$.

When $l = 1$ and $d = 200$ (this case remains "hard" so far), similarly, the Euclidean norm of the first vector is $4q$, and $\|\mathbf{e}\|$ is upper-bounded by $q\sqrt{d+1} \approx 14q$. With similar computation, we can know that the expected norm is around $8q$. This means that many linear combinations of the first d rows will have smaller Euclidean norm than the difference vector $\mathbf{e}$. In other words, there are exponentially many lattice vectors that are closer to the target vector than the hidden vector, thus making decoding extremely difficult.

3.3.2 Modeling Lattice Attacks

Following the idea of [AFG14a], we consider lattice attacks on (EC)DSA as Unique-SVP instances. In [GN08], it is concluded that given a lattice reduction algorithm which we assume to be characterised by a root Hermite factor δ_0 and a n-dimensional lattice $\mathcal{L}$, the algorithm will be successful in disclosing a shortest non-zero vector with "high probability" when $\frac{\lambda_2}{\lambda_1} \geq \tau \cdot {\delta_0}^n$ (we call $\frac{\lambda_2}{\lambda_1}$ the gap), where τ is a constant depending both on the nature of the lattices involved and lattice reduction algorithm being used. However, in [GN08], they do not explain what "high probability" means.

Therefore, in some subsequent work [AFG14a], the success rate is fixed to some number (10 percent, for example) and the dimension n is taken as the smallest possible in practice in order to achieve the same success rate.

Here we slightly change the model such that τ is not a constant, but a function $\tau_n = \frac{k}{\log(n)}$ (k is some constant) on the dimension n. Besides, we choose BKZ–30 as the lattice reduction algorithm and fix the success rate to be 20%. In the context of this section, the modulus q is 160-bit.

Recall that the lattice we construct is:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 2^{l+1} \end{pmatrix}$$

so the lattice dimension $n = d + 1$, where d is the number of signatures being used. As before, we denote the difference vector between the target vector and the hidden lattice vector as $\mathbf{e}$. Besides, we are using Kannan's embedding method to perform lattice attacks on a larger lattice:

$$C = \begin{pmatrix} B & 0 \\ \mathbf{v} & q \end{pmatrix}.$$

Regarded as a Unique-SVP instance, the success rate of the attack crucially depends on the ratio $\frac{\lambda_2'}{\lambda_1'}$, where λ_1', λ_2' are the first and second minimum of the embedded lattice $\mathcal{L}(C)$. According to the relation between $\mathcal{L}(B)$ (lattice spanned by the matrix B) and $\mathcal{L}(C)$ (lattice spanned by the matrix C), $\lambda_1' \approx \|\mathbf{e}\|$ and $\lambda_2' \approx \lambda_1$, where λ_1 is the first minimum of the original lattice $\mathcal{L}(B)$. Therefore, the success rate of lattice attacks increases as the ratio $\frac{\lambda_1}{\|\mathbf{e}\|}$ increases.

As previously mentioned, we assume that in order to achieve 20% success rate, the requirement is

$$\text{gap} = \frac{\lambda_1}{\|\mathbf{e}\|} \geq \frac{k}{\log(d+1)} \cdot {\delta_0}^{d+1},$$

where δ_0 is the root Hermite factor and k is some constant. Before proceeding, we have to determine the root Hermite factor as well as the number of signatures d. First we do some experiments to determine the root hermite factor δ_0 for BKZ–30 on this type of lattice (for HNP attack). After doing numerous experiments, we determine that $\delta_0 \approx 1.01$ for BKZ–30. In addition, we take d as the binary length of the modulus $qlen$ divided by the leakage l. Intuitively and information theoretically, one HNP inequality with l-bit leakage contains l bits of information, so in order to recover the secret key α that has $qlen$ bits, at least $\frac{qlen}{l}$ inequalities are necessary.

Table 3.3: Experimental result: gap needed to achieve $\geq 20\%$ success rate.

Leakage l	Signatures d	Gap $\lambda_1/\|\mathbf{e}\|$	Success rate
3	54	0.93	20/100
4	40	0.91	25/100
5	32	0.88	20/100
6	27	0.87	21/100
7	23	0.86	25/100
8	20	0.85	23/100

Table 3.3 shows the experimental results of the gap ($\frac{\lambda_1}{\|\mathbf{e}\|}$) that is necessary to achieve 20% success rate for 160-bit (EC)DSA. Now we do a linear regression to determine the constant k. After carrying out the regression depicted in Figure 3.1, we find that $k \approx 3.11$.

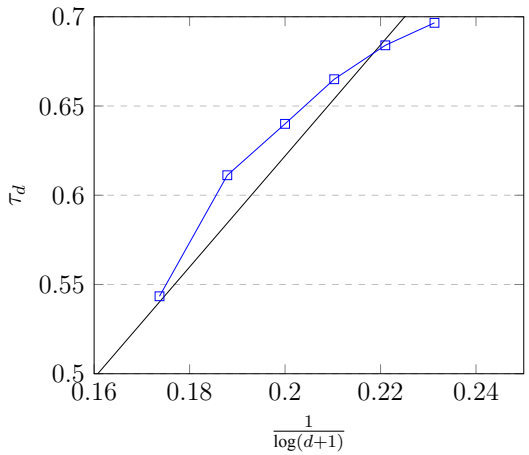


Figure 3.1: Linear regression to estimate the constant k.

Now we estimate the computation cost for the "borderline" case 160-bit (EC)DSA with 2-bit nonce leakage. The dimension $d = \frac{160}{2} = 80$, and the requirement is

$$\text{gap} = \frac{\lambda_1}{\|\mathbf{e}\|} \geq \frac{k}{\log(d+1)} \cdot {\delta_0}^{d+1} = \frac{3.11}{\log(81)} \cdot 1.01^{81} \approx 1.098.$$

According to Gaussian Heuristic, we have

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{d+1}{2\pi e}} \text{vol}(\mathcal{L})^{1/(d+1)} = \sqrt{\frac{81}{2\pi e}} \text{vol}(\mathcal{L})^{1/81}.$$

The expected length of $\mathbf{e}$ is roughly $\sqrt{\frac{81}{3}}q$, so we have

$$\frac{\sqrt{\frac{81}{2\pi e}}\text{vol}(\mathcal{L})^{1/81}}{\sqrt{\frac{81}{3}}q} \geq 1.098,$$

which is equivalent to

$$\text{vol}(\mathcal{L}) \geq 2.61^{81} \cdot q^{81},$$

but the real volume of the lattice is $8^{81} \cdot q^{80}$, so the ratio between them is

$$\frac{2.61^{81} \cdot q^{81}}{8^{81} \cdot q^{80}} \approx \frac{q}{2^{130}} \approx 2^{30}.$$

This means that if we could increase the volume of the lattice by 2^{30} times and keep $\|\mathbf{e}\|$ unchanged, then we have 20% success rate for 160-bit modulus with 2-bit nonce leakage. The number 2^{30} somewhat shows the magnitude of computation cost for 2-bit nonce leakage case.

3.3.3 One Intuitive Idea to Improve the Attacks

The direct idea is to increase the gap $\frac{\lambda_1(\mathcal{L})}{\|\mathbf{e}\|}$. Since $\lambda_1(\mathcal{L}) \approx \sqrt{\frac{d+1}{2\pi e}}\text{vol}(\mathcal{L})^{1/(d+1)}$, we could increase the volume of the lattice while keeping $\|\mathbf{e}\|$ almost unchanged. Our attack is directly based on this idea. In later sections, we will show that by brute-forcing some bits of the secret key (or the nonces), we could modify the original lattice and increase the volume of the lattice, while $\|\mathbf{e}\|$ is almost unchanged. Thus, according to the property of Unique-SVP, we will have significantly better success rate.

3.4 Guessing Bits of Secret Key

In our context, we are considering those "borderline" cases, so in this section, the modulus q has 160 bits and the nonce leakage $l = 2$ (for other moduli, it is similar). In standard lattice attacks, either we find the secret key or get nothing. Even if we set the secret key having only 10 bits, it still does not make lattice attacks any easier (of course, if it has only 10 bits, then we could brute-force the secret key, but it is irrelevant here, since we only care about lattice attacks). Therefore, it is somewhat believed that partial information of the secret key do not help the attack. Perhaps surprisingly, we find that the length of the secret key is closely related to the difficulty

of the attack. Take 160-bit (EC)DSA with 2-bit leakage for instance, if we assume that the secret key has less than 60 bits, we can modify the original lattice and make the attack very easy.

Recall that the HNP inequality is $|\alpha t_i - u_i|_q < q/2^l$ $(i = 1, \ldots, d)$ and the lattice we construct is:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 1 \end{pmatrix}.$$

The target vector is $(2^{l+1}u_1 + q, 2^{l+1}u_2 + q, \ldots, 2^{l+1}u_d + q, 0)$, and the hidden lattice vector is $(\alpha 2^{l+1}t_1 + c_1 2^{l+1}q, \alpha 2^{l+1}t_2 + c_2 2^{l+1}q, \ldots, \alpha 2^{l+1}t_d + c_d 2^{l+1}q, \alpha)$. Again we denote the difference vector between them as $\mathbf{e}$ and we already know that each coefficient of $\mathbf{e}$ is less than q, so $\|\mathbf{e}\| < q\sqrt{d+1}$. As we have discussed in the previous section, in order to improve the success rate of lattice attacks, one direct idea is to increase the volume of the lattice while keeping $\|\mathbf{e}\|$ almost unchanged. For instance, we could modify the lattice as

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 2^{100} \end{pmatrix}.$$

In this way, we increase the volume of the lattice by 2^{100} times, but the problem is that the hidden lattice vector will not be close to the target vector anymore, because the hidden lattice vector is $(\alpha 2^{l+1}t_1 + c_1 2^{l+1}q, \alpha 2^{l+1}t_2 + c_2 2^{l+1}q, \ldots, \alpha 2^{l+1}t_d + c_d 2^{l+1}q, 2^{100}\alpha)$, and the last coefficient of $\mathbf{e}$ is very large $(2^{100}\alpha)$, thus making the modification meaningless.

However, if we assume that the secret key has less than 60 bits, then $2^{100}\alpha$ is still upper-bounded by $2^{160} \approx q$, so this means $\|\mathbf{e}\|$ keeps almost unchanged, and we have increased the volume of the lattice by 2^{100} times, thus making the success probability significantly better. We carry out some simulation experiments and find that if the secret key only has 60 bits for 160-bit (EC)DSA with 2-bit nonce leakage, after modifying the lattice as the above matrix B, we can recover the secret key in just one BKZ–20 operation with 100% success rate, so this becomes almost trivial.

This observation leads to the following attack. First write the secret key in the following format

$$\alpha = \alpha_1 \cdot 2^c + \alpha_2 \ (0 \leq \alpha_2 < 2^c),$$

where c is any arbitrary predetermined integer between 1 and 160. Then α_1 is the $(160-c)$ most significant bits of α and α_2 is the remaining c bits of α. Suppose that we have constructed d HNP inequalities with leakage l

$$|\alpha \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d).$$

Then substitute α with $\alpha_1 \cdot 2^c + \alpha_2$ and we have

$$|\alpha_1 \cdot 2^c \cdot t_i + \alpha_2 \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d).$$

Then set

$$\begin{aligned} t'_i &= 2^c \cdot t_i, \\ u'_i &= -\alpha_2 \cdot t_i + u_i, \end{aligned}$$

so we have new HNP inequalities for t'_i and u'_i:

$$|\alpha_1 \cdot t'_i - u'_i|_q < q/2^l \quad (i = 1, 2, \ldots, d).$$

Then construct the lattice as

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t'_1 & 2^{l+1}t'_2 & \cdots & 2^{l+1}t'_d & 2^c \end{pmatrix}$$

The hidden vector is $(\alpha_1 2^{l+1}t'_1 + c_1 2^{l+1}q, \alpha_1 2^{l+1}t'_2 + c_2 2^{l+1}q, \ldots, \alpha_1 2^{l+1}t'_d + c_d 2^{l+1}q, \alpha_1 2^c)$ and the target vector is $(2^{l+1}u'_1 + q, 2^{l+1}u'_2 + q, \ldots, 2^{l+1}u'_d + q, 0)$. Now we have increased the volume of the lattice by 2^c times while keeping $\|\mathbf{e}\|$ almost unchanged, since $\alpha_1 2^c$ is upper bounded by q. Of course we do not know the value of α_2, but we can enumerate α_2 from 0 to 2^c, so this is a trade-off: we increase the volume of the lattice by 2^c times (thus making the attack easier) at the cost of 2^c enumerations. We formalize the attack as the following steps:

- Step 1: Determine the integer constant c (it depends on how much computation cost we want to pay).
- Step 2: Collect d signatures and construct t_i, u_i as previously defined ($i = 1, 2, \ldots, d$).
- Step 3: Enumerate α_2 from 0 to 2^c:
  - Construct the corresponding HNP instance for α_1.

– Solve the new HNP instance by Kannan's embedding method.

With this method, we are able to attack those "borderline" cases: 160-bit (EC)DSA with 2-bit nonce leakage, 256-bit (EC)DSA with 3-bit nonce leakage, 384-bit (EC)DSA with 4-bit nonce leakage. For more detail, see the section of experimental results.

One typical question for this attack would be: *What is the difference between our approach and directly applying BKZ with larger block size?* We make a comparison:

- In some sense, our approach has similar effect as directly applying BKZ with larger block size. While BKZ with larger block size outputs lattice basis with smaller root Hermite factor (thus better chance of finding the vector **e**), our approach aims to increase the gap for Unique-SVP and have better success rate due to the property of Unique-SVP.
- Our approach is easy to simulate and control. In simulation experiments, we could assume that we have guessed the correct bits, thus avoding the enumeration, which is difficult to carry out in a short time.
- Our approach can be easily parallelized, because each enumeration of bits is independent. While we do not deny the fact that BKZ with larger block size could also be parallelized, it requires another implementation of the SVP oracle (changing the internal code of fplll library [dt20]), which needs a lot of work.

3.5 Guessing Bits of Nonces

Another similar approach could be made to increase the volume of the lattice. Again in our context, the modulus q has 160 bits, the leakage $l = 2$. For other moduli, it is essentially the same, so we will not discuss it again.

Suppose that now we have d 160-bit (EC)DSA signatures (r_i, s_i) $(i = 1, \ldots, d)$ with 2-bit nonce leakage and computed $t_i = |r \cdot 2^{-2} s^{-1}|_q$ and $u_i = |-h(m) \cdot 2^{-2} s^{-1}|_q$ as in previous sections, so the nonce $k_i = 2^2 b_i$ where b_i is some integer. We can guess the third least significant bit of the nonce, thus constructing a HNP inequality with 3-bit leakage with probability $\frac{1}{2}$. If the third bit is zero, then

$$k_i = 2^3 b_i',$$

and we set:

$$t_i' = |r \cdot 2^{-3} s^{-1}|_q \quad \text{and} \quad u_i' = |-h(m) \cdot 2^{-3} s^{-1}|_q.$$

If the third bit is 1, we have:

$$\begin{aligned} k_i &= 2^3 b_i' + 2^2 \\ \alpha r s^{-1} &\equiv 2^3 b_i' + 2^2 - h(m) s^{-1} \pmod q \\ \alpha r s^{-1} 2^{-3} &\equiv b_i' + 2^{-1} - h(m) s^{-1} 2^{-3} \pmod q, \end{aligned}$$

and we then set:

$$t'_i = |r \cdot 2^{-3} s^{-1}|_q \quad \text{and} \quad u'_i = |2^{-1} - h(m) \cdot 2^{-3} s^{-1}|_q.$$

Note that here 2^{-1} means the inverse of 2 mod q, not the fractional number $\frac{1}{2}$. Although we do not know whether the third least significant bit is 0 or 1, by trying these two new settings of t'_i and u'_i, we are essentially guessing the third bit and construct t'_i and u'_i with 3-bit leakage, of which the success probability is $\frac{1}{2}$. Recall that typically, for 2-bit leakage, we need about 90 signatures to perform the attack. Thus the lattice basis is the following matrix B.

$$B = \begin{pmatrix} 8q & 0 & \cdots & 0 & 0 \\ 0 & 8q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 8q & 0 \\ 8t_1 & 8t_2 & \cdots & 8t_{90} & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 16q & 0 & \cdots & 0 & 0 \\ 0 & 16q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 16q & 0 \\ 16t'_1 & 16t'_2 & \cdots & 16t'_{90} & 1 \end{pmatrix}.$$

By guessing one more bit for all the signatures, we can construct the above matrix C with all the inequalities having 3-bit leakage. Of course, with this new matrix, we could attack 160-bit (EC)DSA easily, since we know that for 3-bit leakage, standard lattice attacks work well. However, we are paying a price of 2^{90} for guessing one more bit for all the signatures, which is unacceptable. In order to avoid the huge computation, instead of guessing one more bit for all the signatures, we could guess one more bit for part of the signatures, thus constructing a hybrid lattice. For instance, we can guess one more bit for 20 out of the 90 signatures and keep the other 70 signatures unchanged as follows:

$$D = \begin{pmatrix} 16q & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ & \vdots & 16q & 0 & & \vdots & \vdots \\ 0 & \cdots & 0 & 8q & \cdots & 0 & 0 \\ 0 & \cdots & 0 & \vdots & \cdots & \vdots & \vdots \\ 16t'_1 & \cdots & 16t'_{20} & 8t_{21} & \cdots & 8t_{90} & 1 \end{pmatrix}$$

Now we have increased the volume of the lattice by 2^{20} times and perform the lattice attacks on the new matrix at the cost of 2^{20} operations for guessing bits.

This approach can be summarised as the following steps:

- Step 1: Determine integer constant k and collect d signatures (r_i, s_i) ($i = 1, \cdots, d$), and construct t_i and u_i with the original 2-bit leakage.
- Step 2: For k of them, guess and enumerate the third least significant bit of nonces and construct t'_i and u'_i with 3-bit leakage. For all the other signatures, keep t_i and u_i unchanged.

- Step 3: Construct the hybrid lattice and use Kannan's embedding method to find the secret key (for lattice reduction, we use BKZ–30). If failed, go back to step 2.

Under worst circumstances, we have to perform 2^k times step 2 and 3, since there are 2^k possibilities of the third bits of the nonces.

With this method, we are able to attack those "borderline" cases: 160-bit (EC)DSA with 2-bit nonce leakage, 256-bit (EC)DSA with 3-bit nonce leakage, 384-bit (EC)DSA with 4-bit nonce leakage. See the section of experimental results. Here we make a comparison with the approach in Section 3.4. Generally, the approach in Section 3.4 performs better than the approach in this section. The lattice attack on HNP essentially amounts to decoding a lattice point in a hypercube. When we guess bits of some of the signature nonces, we reduce the length of certain sides of this hypercube. On the contrary, when we guess bits of the secret key, we uniformly shrink the hypercube. For the same exhaustive search cost, the two decoding regions have the same volume, but the average (squared) error length is smaller in the second case.

3.6 Utilizing More Data to Improve Lattice Attacks

In 2000, Bleichenbacher presented a purely statistical attack technique against biased nonces at the IEEE P1363 meeting [Ble00]. The main idea of Bleichenbacher's attack is to define a bias function and search for a candidate value that is near the secret key, thus finding many MSBs of the secret key. An advantage of Bleichenbacher attack is that it can deal with small biases in principle at the cost of using many signatures as input. There is a question in the community (mentioned by cryptanalysis experts on different occasions, e.g., ECC-17 by Tibouchi [Tib17], Lattice Camp-20 by Heninger [Hen20]): *Is it possible to improve lattice attacks with many more signatures?* We give a solution to this question and again we are in the context of 160-bit modulus with 2-bit nonce leakage.

3.6.1 From Bleichenbacher to Lattice

Motivated by Bleichenbacher attack, similar ideas could be applied to lattice attacks. Suppose that we have d HNP inequalities with l-bit leakage

$$|\alpha \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d),$$

and write the secret key α as

$$\alpha = \alpha_1 \cdot 2^c + \alpha_2 \quad (0 \leq \alpha_2 < 2^c).$$

Where α_1 is the $(160-c)$ MSBs of α and α_2 is the remaining LSBs. If t_i ($i = 1, 2, \ldots, d$) is small enough, $\alpha_2 \cdot t_i$ ($i = 1, 2, \ldots, d$) will be a very small perturbation compared with $q/2^l$. This means that with high probability, $\alpha_1 \cdot 2^c$ will satisfy all the d inequalities:

$$|\alpha_1 \cdot 2^c \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d).$$

Then construct the lattice as:

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^c \cdot 2^{l+1}t_1 & 2^c \cdot 2^{l+1}t_2 & \cdots & 2^c \cdot 2^{l+1}t_d & 2^c \end{pmatrix}$$

In this lattice, $(\alpha_1 2^c \cdot 2^{l+1}t_1 + c_1 2^{l+1}q, \alpha_1 2^c \cdot 2^{l+1}t_2 + c_2 2^{l+1}q, \ldots, \alpha_1 2^c \cdot 2^{l+1}t_d + c_d 2^{l+1}q, \alpha_1 2^c)$ will be the hidden lattice vector. The advantage that we get is that the volume of the lattice is increased by 2^c times, while $\|\mathbf{e}\|$ almost keeps unchanged, thus making the attack much easier. Note that now we do not do enumeration of bits as in previous sections.

This attack can be summarised as the following steps:

- Step 1: Collect signatures (r, s) and set:

$$t = 2^{-l}s^{-1}r \bmod q$$
$$u = -2^{-l}s^{-1}h(m) \bmod q$$

  If t is small enough (smaller than some predetermined bound), then keep the (t, u) pairs, otherwise throw it away.

- Step 2: Keep doing step 1 until we get d pairs (t_i, u_i) ($i = 1, \ldots, d$).

- Step 3: Construct the above lattice and use Kannan's embedding method to do lattice attacks.

- Step 4: Find α_1 which is the $(160-c)$ MSBs of α.

- Step 5: Find the remaining bits of α (for example, we can construct a HNP instance for the remaining bits).

As we previously discussed, once we have recovered many MSBs of α, recovering the remaining bits becomes pretty easy.

3.6.2 A Concrete Example

In order to make it clear, we show a concrete example here. For 160-bit (EC)DSA with 2-bit nonce leakage, we collect t which is less than 2^{140} and write the secret key α as:

$$\alpha = \alpha_1 \cdot 2^{10} + \alpha_2 \ (0 \leq \alpha_2 < 2^{10})$$

and we have:

$$\begin{aligned} |\alpha \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d), \\ |\alpha_1 \cdot 2^{10} \cdot t_i + \alpha_2 \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d). \end{aligned}$$

Since α_2 is less than 2^{10}, $\alpha_2 \cdot t$ is upperbounded by 2^{150}, $q/2^l$ has about 158 bits, so as long as the value $|\alpha \cdot t_i - u_i|_q$ does not lie on the edge of the interval $(0, q/2^l)$ (which happens with small probability), we could just throw the term $\alpha_2 \cdot t_i$ away and have:

$$|\alpha_1 \cdot 2^{10} \cdot t_i - u_i|_q < q/2^l \quad (i = 1, 2, \ldots, d).$$

In order to collect 90 signatures where $t < 2^{140}$, we have to sample about $90 \cdot 2^{20} \approx 2^{27}$ signatures. The advantage is that we increase the volume of the lattice almost for free (considering the fact that sampling a signature is much more efficient than doing one BKZ–30 operation). Therefore, at the cost of using 2^{27} signatures, we are able to attack 160-bit (EC)DSA with 2-bit nonce leakage in just one BKZ–30 operation, which is significantly faster than previous results. For more detail, see the section of experimental results.

3.7 Batch SVP and Kannan Embedding Factor

3.7.1 Batch SVP

In section 3.4 and section 3.5, we have to do 2^c (typically we set $c = 15, 20$) BKZ–30 operations on the following matrices:

$$C = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 & 0 \\ & \vdots & & \vdots & & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 1 & 0 \\ v_1 & v_2 & \cdots & v_d & 0 & q \end{pmatrix}.$$

Write C as

$$C = \begin{pmatrix} B & 0 \\ \mathbf{v} & q \end{pmatrix}.$$

Each time we perform BKZ operations, only the last row of matrix C is changed and B is fixed. One BKZ–30 operation on a 90-dimensional lattice typically takes about 3 minutes with fplll [dt20] library on Sagemath [The20]. If c = 2^{15}, the time complexity will be $2^{15} \cdot 3$ minutes without using multiple cores. Although this is practical time, we could further improve the time complexity. For simplicity, we use LLL as an example here (for BKZ it is similar). In LLL algorithm [LLL$^+$82], there is an index k starting from 1, which represents the row currently being reduced. Besides, there is an exchange condition, and if it is satisfied, two adjacent rows will be exchanged. After exchanging rows and recomputing the Gram–Schmidt norm, size reduction will be performed.

If we consider the process of LLL reduction on the matrix C, essentially it will first reduce the submatrix B, so every time the reduction on B is repeated, which is not necessary. We come up with a simple solution:

- Step 1: BKZ-reduce the submatrix B (preprocessing).
- Step 2: Do Kannan embedding and construct the matrix C.
- Step 3: Do BKZ on the matrix C again.

This actually means that we preprocess the submatrix B. In this way, we save a lot of computation. With this preprocessing, one BKZ–30 operation typically takes several seconds, while the original one takes about 3 minutes.

3.7.2 Kannan Embedding Factor

In our experiments, we observe that lattice attacks on (EC)DSA are very sensitive to the Kannan embedding factor. To the best of our knowledge, there are only a few works that discuss how to choose the factor. For example, in Galbraith's book [Gal12], the embedding factor is set to 1 by default, and in [WAT17], Kannan embedding factor for LWE lattices (very different context) is discussed. Therefore, we give a simple analysis for HNP lattice for completeness. As we can see from Table 3.4, if the factor is either too small or too large, the success rate becomes very low.

Here we give an explanation why this happens. Denote the Kannan embedding factor as γ. For simplicity, we analyze LLL reduction.

Case 1: Kannan embedding factor is too large. Recall that the embedded matrix C is

$$C = \begin{pmatrix} B & 0 \\ \mathbf{v} & \gamma \end{pmatrix}.$$

Table 3.4: Kannan embedding factor test.

Modulus	Leakage	Signatures	Kannan embedding factor	Success rate
160-bit	3	80	q	93/100
160-bit	3	80	$(q-1)/2$	97/100
160-bit	3	80	q^2	5/100
160-bit	3	80	1	0/100

The Gram–Schmidt norm of the last row is γ, and if γ is too large, after LLL reduction on the submatrix B, the exchange condition will not be satisfied, then only one round of size reduction will be performed (reduce the last row from the first (d + 1) rows) and the algorithm terminates. By contrast, if γ is properly valued, the exchange condition will be satisfied and the last row will be exchanged to some other row. Then Gram–Schmidt norm will be recomputed and one round of size reduction will be performed. Typically, the exchange happens many times, so many rounds of size reduction will be performed. Therefore, if γ is too large, the lattice will get much less reduced.

Case 2: Kannan embedding factor is too small. Since the Gram–Schmidt norm of the last row is γ, if the Kannan embedding factor is too small, the Gram–Schmidt norm of the last row will be very small. After exchanging rows, size reduction will be performed. Since the Gram–Schmidt norm is small, when performing size reduction on other rows, many multiples of the target vector **v** will be added to other rows. However, since

$$B = \begin{pmatrix} 2^{l+1}q & 0 & \cdots & 0 & 0 \\ 0 & 2^{l+1}q & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 2^{l+1}q & 0 \\ 2^{l+1}t_1 & 2^{l+1}t_2 & \cdots & 2^{l+1}t_d & 1 \end{pmatrix},$$

what we want is $\alpha \cdot (2^{l+1}t_1, 2^{l+1}t_2, \ldots, 2^{l+1}t_d, 1) - \mathbf{v}$, so we do not want to use the target vector **v** to reduce other vectors. If γ is too small, we will find that all the vectors in the reduced basis will have a very large coefficient of **v**, which is not our goal.

3.8 Gap Between the CVP and SVP Approaches

As mentioned in [JSSS20], we also observe a certain gap between the nearest plane algorithm and Kannan's embedding method. In this attack, Kannan's embedding method always outperforms nearest plane algorithm to some extent.

Table 3.5: A comparison of the CVP and SVP approaches.

Modulus	Leakage	Nearest plane	Kannan's embedding method
160-bit	4-bit	37/100	100/100
160-bit	3-bit	0/100	91/100

As we can see from Table 3.5, for 160-bit (EC)DSA with 4-bit nonce leakage, both approaches work well. However, nearest plane algorithm never succeeds for 3-bit leakage, while Kannan's embedding method works quite well.

Reason for the Gap. Essentially, nearest plane algorithm can be regarded as one round of size reduction in the embedded lattice. Recall the process in the previous section, if the Kannan embedding factor is large enough, nearest plane algorithm will be the same as Kannan embedding, because for the last row of the embedded lattice, the exchange condition will not be satisfied and only one round of size reduction takes place, which is essentially the same as nearest plane. However, if the Kannan embedding factor is properly valued, many exchanges will happen. After one exchange, one round of size reduction will take place, which means that Kannan's embedding method will make the target vector more reduced compared with nearest plane algorithm.

3.9 Experimental Results

In this section, we show the result of our practical experiments. All the experiments are carried out on AMD Ryzen 3970x with Sagemath [The20] and fplll [dt20] library. For lattice reduction algorithms, we are using BKZ–30. The source code is available in [Sun21].

3.9.1 Guessing Bits of Secret Key

As we can see in Figure 3.2, as the number of guessed bits increases, the success rate increases. Take 160-bit (EC)DSA with 2-bit nonce leakage for example, if we guess 15 bits for the secret key, we succeed in recovering the secret key 12 times among 200 experiments. Since we enumerate 15 bits of the secret key, the time complexity is upper bounded by 2^{15} BKZ–30 operations (the expected number is 2^{14}). In this way, we are able to quantify the complexity in terms of BKZ operations. Instead of directly using real-time, the advantage is that it gives us a clear impression of the time complexity and this is independent of the machine being used. Besides, it is easy to estimate the practical attack time. For instance, with Ryzen 3970x and batch

SVP technique described in section 3.7, one BKZ–30 operation on a 90-dimensional lattice takes 40 seconds (on average) on a single core, so the expected time is $\frac{2^{14} \cdot 40s}{32} \approx 10200s$, which is several hours.

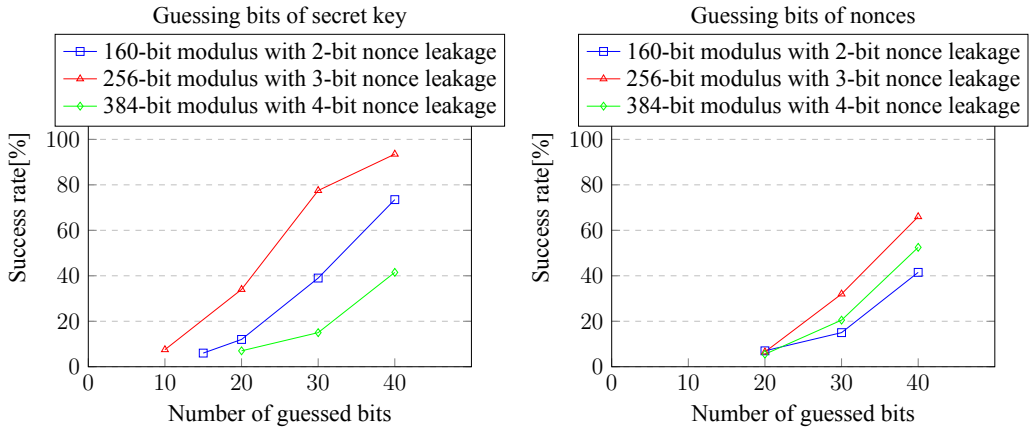


Figure 3.2: Experimental results: guessing bits of the secret key vs. of the nonces.

3.9.2 Guessing Bits of Nonces

Similarly, for 160-bit (EC)DSA with 2-bit nonce leakage, if guessing 1 more bit for 20 of the 90 signatures, we succeed in recovering the secret key 14 times out of 200 experiments, so the time complexity is 2^{20} BKZ–30 operations. Actually, we could even estimate the time complexity for 1-bit nonce leakage. What we could do is to guess 2 more bits for 20 of the signatures and guess 1 more bit for the other 70 signatures, so the time complexity is $4^{20} \cdot 2^{70} = 2^{110}$ BKZ–30 operations. Although this is not practical (thus not so meaningful), it is an estimate of computation cost for 1-bit leakage.

3.9.3 Improving Lattice Attacks with More Data

Recall that in Section 3.6, we discussed that for one HNP inequality $|\alpha t - u|_q < q/2^l$, if we get small t, then we can construct a lattice that has larger volume. In our experiments, summarized in Table 3.6, we find that for 160, 256, 384-bit modulus q, if t has less than 140, 226, 344 bits respectively, we can perform the attack. Take 160-bit modulus for example, in order to get 90 inequalities where all the $t \leq 2^{140}$, we have to sample $2^{20} \cdot 90 \approx 2^{27}$ signatures. This may seem too many in practical setting, but the advantage is that we could recover about 150 MSBs of the secret key in just one BKZ-30 operation.

Table 3.6: Utilizing more data to improve lattice attacks.

Modulus	Leakage	Upper bound on t	Signatures	Time complexity	Success rate
160-bit	2-bit	2^{140}	2^{27}	1 BKZ-30	30/200
256-bit	3-bit	2^{226}	2^{37}	1 BKZ-30	27/200
384-bit	4-bit	2^{344}	2^{47}	1 BKZ-30	62/200

3.9.4 Experiments on the TPM–FAIL Dataset

We also carry out experiments on the TPM–FAIL [MSEH20] dataset (256-bit ECDSA). The first row of the dataset contains the public key and the message being signed. Each of the other rows contains (r, s) and t, where (r, s) is the signature and t is the signing time. One typical way to perform the attack is:

- Collect N signatures.
- Choose d out of the N signatures, whose signing time is the fastest.
- For each of the d signatures, assign leakage l.
- Construct HNP inequalities and perform lattice attacks.

For 256-bit modulus, if setting $l = 3$, typically $d \approx 90$. In [MSEH20], the authors use about 40000 signatures and in Minerva [JSSS20], a new technique of geometric assignment of leakage is proposed: assign half of the d signatures with leakage $l = 3$, one fourth of them having leakage $l = 4$, and so on. In our experiments, we combine these techniques with our method of guessing bits of the secret key and come up with the following attack:

- Randomly collect 800 signatures.
- Choose 90 out of the 800 signatures, whose signing time is the fastest.
- Geometrically assign the leakage l.
- Guess and enumerate some LSBs of the secret key and perform lattice attacks described in section 3.4.

We do 100 experiments and succeed 3 times. In this way, with only 800 signatures available, we are able to recover the secret key for TPM–FAIL dataset. For the most part of this work, we are in a setting where there is no noise in the sense that leakage is assigned correctly for each signature. However, this is not the case in general in practice. If the number of signatures is enough, it is easy to assign the leakage

correctly with high probability, but if $N = 800$, it is unavoidable that some of the assignment are wrong, which is somewhat annoying and makes the success rate very low. There are many robust techniques in Minerva [JSSS20] for dealing with noise, which are very important contribution of that paper. For example, the *random subset technique* in Minerva could be utilized: instead of choosing d out of N signatures, we could choose $1.5d$ signatures and collect a random subset having d elements. Besides, the *CVP + flip technique* can be applied: change **u** to correct errors (this part can even be generalized with our nonce guessing technique by flipping more bits). Considering that our work is largely orthogonal and complementary to Minerva and we only use BKZ–30 (which could be replaced with stronger lattice reduction algorithms, e.g., [CN11, AWHT16, ADH$^+$19, EK20, KEF21]), it is fair to say that our approaches help improving the attack.

Chapter 4

Constructing Efficient and Secure Lattice-based Signatures

This chapter is based on joint work with Thi Thu Quyen Nguyen, Thomas Espitau, Alexandre Wallet and my supervisors.

4.1 Introduction

4.1.1 Hash-and-sign lattice-based signatures

From GGH to Falcon. Falcon [PFH+22] is one of the three signature schemes already selected for standardization in the NIST postquantum competition. It represents the state of the art in *hash-and-sign* lattice-based signatures, one of the two main paradigms for constructing lattice-based signatures alongside Lyubashevsky's Fiat–Shamir with aborts [Lyu09, Lyu12] (which is also represented among the final selected candidates of the NIST competition in the form of Dilithium [LDK+22]).

This makes Falcon the culmination of a long line of research in constructing signature schemes from *lattice trapdoors*. The basic idea, which dates back to the late 1990s with the GGH [GGH97] and NTRUSign [HHP+03] signature schemes, is to use as signing key a "good" basis (the *trapdoor*) of a certain lattice allowing to approximate the closest vector problem to a good factor, and as the verification key a "bad" basis which allows to test membership but not decode large errors. The signature algorithm then hashes a given message to a vector in the ambient space of the lattice, and uses the the trapdoor to find a relatively close lattice point to that vector. The difference is the signature, which is verified by checking that it is small and that its difference with the hashed vector belongs to the lattice.

The GGH scheme, as well as several successive variants of NTRUSign, were eventually broken by statistical attacks [GS02, NR06, DN12]: it turned out that sig-

natures would reveal partial information about the secret trapdoor, that could then be progressively recovered by an attacker. This problem was finally solved in 2008, when Gentry, Peikert and Vaikuntanathan (GPV) [GPV08] showed how to use Gaussian sampling in the lattice in order to guarantee that signatures would reveal no information about the trapdoor.

GPV signatures over NTRU lattices. In order to instantiate the GPV framework efficiently in practice, one then needs lattices with compact representation and efficiently computable trapdoors, which has so far been achieved using module lattices over rings—in fact, mostly rank-2 modules over cyclotomic rings, exactly corresponding to NTRU lattices (although higher rank modules, namely ModNTRU lattices, have been shown to be usable as well in certain in certain ranges of parameters [CPS+20]). This was first carried out by Ducas, Lyubashevsky and Prest (DLP) [DLP14], who analyzed trapdoor generation for power-of-two cyclotomic ring NTRU lattices and constructed corresponding GPV-style signatures. DLP signatures are compact, but the signing algorithm is rather slow: quadratic in the dimension $2d$ of the lattice. This is because the lattice Gaussian sampling algorithm that forms the core of its signing procedure (namely Klein–GPV sampling, in essence a randomized version of Babai's nearest plane algorithm for approximate CVP) cannot directly take advantage of the algebraic structure of the lattice, and thus operates on the full $(2d) \times (2d)$ matrix of the lattice basis as well as its Gram–Schmidt orthogonalization.

Falcon is a direct descendent of the DLP scheme, that replaces the generic, quadratic complexity Klein–GPV sampler in signature generation by an efficient, quasilinear compexity lattice Gaussian sampler that *does* take advantage of the ring structure. Specifically, that new algorithm is constructed by randomizing the Fast Fourier Orthogonalization (FFO) algorithm of Ducas and Prest [DP15a], and operates in a tree-like fashion traversing the subfields of the power-of-two cyclotomic field over which the NTRU lattice is defined. This makes Falcon particularly attractive in various ways: it offers particularly compact signatures and keys (providing the best bandwidth requirements of all signature schemes in the NIST competition), achieves high security levels in relatively small lattice dimensions, and has both fast signing and very efficient verification speeds.

However, the FFO-based Gaussian sampler is also the source of Falcon's main drawbacks: it is a really contrived algorithm that is difficult to implement correctly, parallelize or protect against side-channels. It is also really difficult to adapt to other rings than power-of-two cyclotomics, which drastically limits Falcon's versatility in terms of parameter selection: in fact, recent versions of Falcon in the NIST competition only target either the lowest NIST security level (using cyclotomic fields of dimension 512) or the highest (using fields of dimension 1024) and nothing inbe-

tween.[1]

4.1.2 The hybrid sampler and Mitaka

The Peikert and hybrid samplers. After the publication of the DLP paper, Ducas and Prest explored and analyzed other approaches for lattice Gaussian sampling over NTRU lattices, as discussed in depth in Prest's Ph.D. thesis [Pre15], with a view towards overcoming the quadratic complexity of the naive Klein–GPV sampler. While the introduction of the FFO sampler was the final step of that exploration, they also considered two other major approaches along the way, which also achieve quasilinear complexity (see also [DP15b]).

The first approach was not actually novel: it was the ring version of Peikert's lattice Gaussian sampler [Pei10], which is the randomization of the Babai rounding algorithm for approximate CVP, just like Klein–GPV is the randomization of Babai's nearest plane. For NTRU lattices, this algorithm consists of independent one-dimensional Gaussian samplings for each vector component (hence a linear number in total), as well as 2×2 matrix-vector products over the ring, amounting to a constant number of ring multiplications, that are all quasilinear when using FFT-based fast arithmetic. Thus, Peikert's sampler for NTRU lattices is quasilinear as required. However, Ducas and Prest analyzed the *quality* of NTRU trapdoors (generated in the same way as DLP) with respect to Peikert's sampler, and found that it was much worse than for Klein–GPV, both concretely and asymptotically. In other words, for the same choice of parameters, it would reduce security considerably to instantiate DLP with Peikert's sampler instead of Klein–GPV (and to recover the same security, a large increase in the dimension of the underlying ring, and hence the size of keys and signatures, would be required).

As a kind of middle ground between Peikert (fast but less secure) and Klein–GPV (secure but much slower), they introduced as a second approach the *hybrid sampler*, which uses the same structure as Klein–GPV (a randomized nearest plane algorithm) but over the ring instead of over $\mathbb{Z}$. In the rank-2 case of NTRU, this reduces to just two "nearest plane" iterations consisting of Gaussian sampling over the ring, which is itself carried out using Peikert's sampler with respect to a short basis of the ring. This algorithm remains quasilinear, but achieves a significantly better quality than Peikert for DLP-style NTRU trapdoors, although not as good as Klein–GPV. Concretely, for those NTRU trapdoors over the cyclotomic ring of dimension 512 (resp. 1024), signatures instantiated with the hybrid sampler achieve a little over 80 bits (resp. 200 bits) of classical CoreSVP security, compared to over 120 bits (resp.

[1] The earliest version of the Falcon specification [PFH+17] also included an intermediate parameter set of dimension 768, but the corresponding algorithms were so complicated that it was eventually dropped.

280 bits) for Klein–GPV.

Pros and cons of hybrid vs. FFO. This substantial security loss is presumably the main reason that led to the hybrid sampler being abandoned in favor of the FFO sampler (which achieves the same quality as Klein–GPV but with quasilinear complexity) in the Falcon scheme. Indeed, security aside, the hybrid sampler has a number of advantages compared to the FFO sampler of Falcon: it is considerably simpler to implement, somewhat more efficient in equal dimension, easily parallelizable and less difficult to protect against side-channels; it also has an online-offline structure that can be convenient for certain applications, and it is easier to instantiate over non power-of-two cyclotomics, making it easier to reach intermediate security levels.

For these reasons, the use of the hybrid sampler to instantiate signatures over NTRU lattices was recently revisted by Espitau et al. as part of their proposed scheme Mitaka [EFG+22]. One of the key contributions of that paper is an optimization of trapdoor generation for the hybrid sampler that mitigates the security loss by making it possible to construct better quality trapdoor in reasonable time. Combined with the various advantages of the hybrid sampler, this allows the authors of Mitaka to achieve a trade-off between simplicity and security that they argue can be more attractive than Falcon. However, despite their efforts, Mitaka remains substantially less secure than Falcon in equal dimension (it loses over 20 bits of classical CoreSVP security over rings of dimension 512, and over 50 bits over rings of dimension 1024), with a much slower and more contrived key generation algorithm as well. In particular, Mitaka falls short of NIST security level I in dimension 512 and of level V in dimension 1024, making it less than ideal from the standpoint of parameter selection.

4.1.3 Contributions and technical overview of this work

In this work, we introduce a novel trapdoor generation technique for Prest's hybrid sampler that solves the issues faced by Mitaka in a natural and elegant fashion. Our technique gives rise to a much simpler and faster key generation algorithm than Mitaka's (achieving similar speeds to Falcon), and it is able to comfortably generate trapdoors reaching the same NIST security levels as Falcon. It can also be easily adapted to rings of intermediate dimensions, in order to support the same versatility as Mitaka in terms of parameter selection (just with better security). All in all, this new technique achieves in some sense the best of both worlds between Falcon and Mitaka.

NTRU trapdoors and their quality. In order to give a overview of the technical ideas involved, we need to recall a few facts about NTRU trapdoors and their quality with respect to the Klein–GPV and hybrid samplers. For simplicity, we concentrate

on the special case of power-of-two cyclotomic rings $\mathscr{R} = \mathbb{Z}[x]/(x^d + 1)$. Over such a ring, an NTRU lattice is simply a full-rank submodule lattice of $\mathscr{R}^2$ generated by the columns of a matrix of the form:

$$\mathbf{B}_h = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix}$$

for some rational prime number q and some ring element h coprime to q. Note that this can also be described as a lattice of pairs $(u, v) \in \mathscr{R}^2$ such that $uh - v = 0 \bmod q$.

A trapdoor for this lattice is a relatively short basis:

$$\mathbf{B}_{f,g} = \begin{bmatrix} f & F \\ g & G \end{bmatrix}$$

where the basis vectors (f, g) and (F, G) are not much larger than the normalized volume $\sqrt{\det \mathbf{B}_h} = \sqrt{q}$ of the lattice. Since those vectors belong to the lattice, we have in particular that $g/f = G/F = h \bmod q$. Moreover, since the determinants are equal up to a unit of $\mathscr{R}$, we can impose without loss of generality that $fG - gF = q$.

Using the trapdoor $\mathbf{B}_{f,g}$, lattice Gaussian samplers are able to output lattice vectors following a Gaussian distribution on the lattice of standard deviation[2] to output a small multiple $\alpha\sqrt{q}$ of the normalized volume $\sqrt{q}$. The factor α is the *quality*, and depends both on the trapdoor and on the sampler itself. The lower the quality, the better the trapdoor, and the higher the security level of the resulting signature scheme. For the Klein–GPV sampler, one can show that the quality α is ($1/\sqrt{q}$ times) the maximum norm of a vector in the Gram–Schmidt orthogonalization of the basis $\mathbf{B}_{f,g}$ regarded as a $(2d) \times (2d)$ matrix over $\mathbb{Z}$, whereas for the hybrid sampler, it is similar but with the Gram–Schmidt orthogonalization over $\mathscr{R}$ itself.

Those quantities admit a simple expression in terms of the *embeddings* of the ring elements f and g. Recall that the embeddings are the d ring homomorphisms $\varphi_i \colon \mathscr{R} \to \mathbb{C}$; when elements of $\mathscr{R}$ are seen as polynomials, these embeddings are simply the evaluation morphisms $\varphi_i(u) = u(\zeta_i)$ where the ζ_i's are the d primitive $2d$-th roots of unity in $\mathbb{C}$. Then, quality of the basis $\mathbf{B}_{f,g}$ with respect to the Klein–GPV sampler admits the following simple expression:

$$\alpha_{\text{GPV}} = \max\left(\frac{1}{d}\sum_{i=1}^{d}\frac{|\varphi_i(f)|^2 + |\varphi_i(g)|^2}{q},\ \frac{1}{d}\sum_{i=1}^{d}\frac{q}{|\varphi_i(f)|^2 + |\varphi_i(g)|^2}\right).$$

Similarly, the quality with respect to the hybrid sampler is:

$$\alpha_{\text{hybrid}} = \max_{1 \le i \le d}\left(\max\left(\frac{|\varphi_i(f)|^2 + |\varphi_i(g)|^2}{q},\ \frac{q}{|\varphi_i(f)|^2 + |\varphi_i(g)|^2}\right)\right).$$

[2]The actual standard deviation also includes an additional factor (the smoothing parameter of the ring) which we omit in this overview for simplicity's sake.

Note that $|\varphi(f)|^2 + |\varphi_i(g)|^2 = \varphi_i(ff^* + gg^*)$ where the star denotes the complex conjugation automorphism of $\mathscr{R}$ (defined by $x^* = 1/x = -x^{d-1}$). Thus, put differently, one can say that a trapdoor $\mathbf{B}_{f,g}$ achieves quality α or better for the Klein–GPV sampler if and only if the embeddings of $(ff^* + gg^*)/q$ and of its inverse are at most α *on average*, whereas quality α or better is obtained for the hybrid sampler if *all* of the embeddings of these values are at most α. This shows in particular that the quality of a given trapdoor is always at least as good for Klein–GPV as it is for the hybrid sampler, which explains why it may be easier in practice to construct good quality trapdoors for the former than for the latter.

Trapdoor generation in Falcon and Mitaka. Now, the way trapdoors are generated in Falcon is by sampling f and g according to a discrete Gaussian in $\mathscr{R}$ (which can easily be done by sampling the coefficients as discrete Gaussians over $\mathbb{Z}$) so that their expected length is a bit over $\sqrt{q}$, and verifying using the condition above that the quality with respect to the Klein–GPV (or equivalently Falcon's) sampler is $\alpha_{\text{Falcon}} = 1.17$ or better, and restarting otherwise (the value 1.17 here is chosen roughly as small as possible while keeping the number of repetitions relatively small).

The approach to generate trapdoors in Mitaka is similar using the quality formula for the hybrid sampler, and a target quality of $\alpha_{\text{Mitaka}} = 2.04$ in dimension 512 (and slightly increasing as the dimension becomes larger). Doing so directly would take too many repetitions, however, so in fact the candidates for f and g are obtained by linear combinations of smaller Gaussian vectors and by applying Galois automorphisms to generate many candidate vectors (f, g) from a limited number of discrete Gaussian samples. Using that approach, Mitaka achieves the stated quality with a comparable number of discrete Gaussian samples as Falcon; its key generation algorithm is much slower, however, as it has to carry out an exhaustive search on a much larger set of possible candidates.

Our Antrag strategy: annular NTRU trapdoor geneneration. In both Falcon and Mitaka, however, the overall strategy is to generate random-looking candidates (f, g) of plausible length, and repeat until the target quality is reached. In this work, we suggest a completely different strategy that is in some sense much simpler and more natural: just pick the pair (f, g) uniformly at random in the set of vectors that satisfy the desired quality level. We propose and analyze this approach specifically for the hybrid sampler.[3]

[3] One could consider doing so for Klein–GPV as well, but this appears less relevant for two reasons. First, since 1.17 is already quite close to the theoretical optimal quality of 1, and since the number of repetitions in Falcon's key generation is fairly modest, there is not much to gain in the Klein–GPV setting. Second, the space of key candidates has a less elegant geometric description, making it more difficult to sample uniformly in it.

Concretely, yet another way of reformulating the quality condition for the hybrid sampler is to say that the quality is α or better if and only if for all the embeddings φ_i, one has:

$$q/\alpha^2 \leq |\varphi_i(f)|^2 + |\varphi_i(g)|^2 \leq \alpha^2 q.$$

In other words, for each embedding, the pair $\big(|\varphi_i(f)|, |\varphi_i(g)|\big)$ lies in the *annulus* $A\big(\sqrt{q}/\alpha, \alpha\sqrt{q}\big)$ bounded by the circles of radii $\sqrt{q}/\alpha$ and $\alpha\sqrt{q}$—or more precisely, in the *arc* $A^+_\alpha = A^+\big(\sqrt{q}/\alpha, \alpha\sqrt{q}\big)$ of that annulus located in the upper-right quadrant of the plane since those absolute values are non-negative numbers. Our approach is then to sample f and g by their embeddings (i.e., directly in the Fourier domain), and select those embeddings uniformly and independently at random in the desired space. Namely, we sample $d/2$ pairs (x_i, y_i) in the arc of annulus A^+_α, and set the i-th embedding of f (resp. g) to a uniformly random complex number $x_i \cdot e^{i\theta_i}$ of absolute value x_i (resp. of absolute value y_i).

An obvious issue is that the elements f and g constructed in this way will generally not lie in the ring itself: after mapping back to the coefficient domain by Fourier inversion, their coefficients are a priori abitrary real numbers instead of integers. But this is easy to address: we simply round coefficient-wise to obtain an actual ring element.

A second issue is that this rounding step will not necessarily preserve the quality property we started from: the embeddings of the rounded values do not necessarily remain in the correct domain. In fact, the probability that *all* embeddings remain in the correct domain after rounding is very low. But there is again a simple workaround: we just carry out our original continuous sampling in the Fourier domain from a slightly smaller annulus than the target one. Instead of picking the pairs (x_i, y_i) in A^+_α as above, we sample them uniformly in some $A^+(r, R)$ with r slightly larger than $\sqrt{q}/\alpha$ and R slightly smaller than $\alpha\sqrt{q}$. This considerably increases the probability that, after rounding, all of the pairs $\big(|\varphi_i(f)|, |\varphi_i(g)|\big)$ will in fact end up in A^+_α.

And voilà: the description above is essentially a complete trapdoor generation algorithm for the hybrid sampler, that easily reaches the same NIST security levels as Falcon. Concretely, we target $\alpha = 1.17$ in dimension 512 (the same as Falcon) and $\alpha = 1.64$ in dimension 1024 (to obtain the 256 bits of classical CoreSVP security corresponding to NIST level V), and with those numbers, we achieve key generation speeds close to Falcon's, while benefitting of all the advantages of Mitaka in terms of simplicity of implementation, efficiency, parallelizability and so on as far as signing in concerned.

Our contributions. The main contribution of this work is to introduce, analyze and implement the Antrag trapdoor generation algorithm for the hybrid sampler described above.

The analysis includes a heuristic estimate of the success probability of sampling in

the required domain, as well as a discussion of possible attacks on the resulting keys (and even though our security analysis is in a very optimistic model for the attacker, we find no weakness as long as the original sampling domain $A^+(r, R)$ is not chosen to be extremely narrow), and concrete parameters to instantiate a signature scheme.

We also provide, as supplementary material, a full portable C implementation of the corresponding signature scheme based on those of Falcon and Mitaka. In fact, since the C implementation of Mitaka did not include the key generation algorithm, our implementation is the first complete implementation of the corresponding paradigm. This implementation lets us compare the performance of our key generation with Falcon's, and we find that they are quite close.

Although most of the previous discussion was in the context of power-of-two cyclotomics, our approach also extends to other base rings essentially without change. In particular, it is still possible to map candidate continuous random values generated in the Fourier domain to the ring by coefficient-wise rounding (we could consider other decoding techniques, but this one is sufficient for our purposes; it was in fact already used in the original ternary version of Falcon: see [PFH$^+$17, Algorithm 10]). This only changes the distribution of the "rounding error" and hence the success probability slightly, but the analysis carries over easily. It follows that our approach supports the same versatility as Mitaka in terms of parameter settings.

4.2 Preliminaries

Some of the notations are borrowed from the Mitaka paper [EFG$^+$22]. For two real numbers $0 \leq r \leq R$, we denote by $A(r, R)$ the *annulus* limited by radii r and R, i.e. the following subset of the plane $\mathbb{R}^2$: $A(r, R) := \left\{(x, y) \in \mathbb{R}^2 \mid r^2 \leq x^2 + y^2 \leq R^2\right\}$. We also denote by $A^+(r, R)$ the arc of annulus located in the upper-right quadrant of the plane, i.e., $A^+(r, R) := \left\{(x, y) \in A(r, R) \mid x, y \geq 0\right\}$.

If f is a some function over a set S, denote $f(S) = \sum_{s \in S} f(s)$ assuming that this sum is absolutely convergent. $\lfloor \cdot \rceil$ represents the rounding of a real number to its closest integer. For a polynomial f, $\lfloor f \rceil$ represents the polynomial whose each coefficient is rounded to the nearest integer. We use $\mathbf{A}^t$ to represent the transpose matrix of $\mathbf{A}$. Let $Q \in \mathbb{R}^{n \times n}$ be a symmetric matrix. If Q is positive definite, then we write as $Q \succ 0$. i.e. $\mathbf{x}^t Q \mathbf{x} > 0$ for all non-zero $\mathbf{x} \in \mathbb{R}^n$. We also write $Q_1 \succ Q_2$ when $Q_1 - Q_2 \succ 0$. A norm for a vector $\mathbf{x} \in \mathbb{R}^n$ can be defined with the positive definite matrix Q as $||\mathbf{x}||_Q = \sqrt{\mathbf{x}^t Q \mathbf{x}}$. Besides, a bilinear form can be defined as $\langle \mathbf{x}, \mathbf{y} \rangle_Q = \mathbf{x}^t Q \mathbf{y}$. Denote the largest singular value of A as $s_{1,Q}(A) = \max_{x \neq 0} \frac{||\mathbf{Ax}||_\mathbf{Q}}{||\mathbf{x}||_Q}$.

A lattice $\mathscr{L}$ is a discrete additive subgroup in the Euclidean space $\mathbb{R}^m$. A lattice can be generated by one basis $\mathbf{B} \in \mathbb{R}^{m \times d}$ having linearly indepedent columns. d is called the rank of $\mathscr{L}$. The volume of the lattice w.r.t the norm $|| \cdot ||_Q$ is defined $\mathrm{vol}_Q(\mathscr{L}) = \det(\mathbf{B}^t Q \mathbf{B})^{1/2} = |\det(\mathbf{B})| \sqrt{\det(Q)}$ for any basis $\mathbf{B}$.

4.2.1 Cyclotomic fields

Let m be an positive integer, and $d = \phi(m)$ be the degree of the m-th cyclotomic polynomial Φ_m (ϕ is the Euler totient function). Let ζ to be a m-th primitive root of 1. Then for a fixed m, $\mathscr{K} := \mathbb{Q}(\zeta)$ is the cyclotomic field associated with Φ_m, and its ring of algebraic integers is $\mathscr{R} := \mathbb{Z}[\zeta]$. The field automorphism $\zeta \mapsto \zeta^{-1} = \bar{\zeta}$ corresponds to the complex conjugation, and we write f^* the image of f under this automorphism. We have $\mathscr{K} \simeq \mathbb{Q}[x]/(\Phi_m(x))$ and $\mathscr{R} \simeq \mathbb{Z}[x]/(\Phi_m(x))$, and both are contained in $\mathscr{K}_\mathbb{R} := \mathscr{K} \otimes \mathbb{R} = \mathbb{R}[x]/(\Phi_m(x))$. Each $f = \sum_{i=0}^{d-1} f_i \zeta^i \in \mathscr{K}_\mathbb{R}$ can be identified with its coefficient vector $(f_0, \cdots, f_{d-1}) \in \mathbb{R}^d$. The adjoint operation extends naturally to $\mathscr{K}_\mathbb{R}$, and $\mathscr{K}_\mathbb{R}^+$ is the subspace of elements satisfying $f^* = f$.

The cyclotomic field $\mathscr{K}$ has d complex field embeddings $\varphi_i : \mathscr{K} \to \mathbb{C}$, where each embedding maps f to its evaluations at all the primitive roots of unity ζ^k where $\gcd(k, m) = 1$. This is usually called the canonical embedding $\varphi(f) := (\varphi_1(f), \ldots, \varphi_d(f))$. The embedding can also naturally applied to $\mathscr{K}_\mathbb{R}$ and maps to the space $\mathcal{H} = \{v \in \mathbb{C}^d : v_i = \overline{v_{d/2+i}}, 1 \leq i \leq d/2\}$. According to the properties of embeddings, $\varphi(fg) = (\varphi_i(f)\varphi_i(g))_{0<i\leq d}$. When needed, this embedding extends entry-wise to vectors or matrices over $\mathscr{K}_\mathbb{R}$. We let $\mathscr{K}_\mathbb{R}^{++}$ be the subset of $\mathscr{K}_\mathbb{R}^+$ whose canonical embedding has all positive coordinates. We have a partial ordering over $\mathscr{K}_\mathbb{R}^+$ by $f \succ g$ if and only if $f - g \in \mathscr{K}_\mathbb{R}^{++}$. We can also equip the algebra $\mathscr{K}_\mathbb{R}$ with a norm $N_\mathscr{K}(x) = \prod_i \varphi(x)$, which extends the standard field norm.

The next technical lemma is useful in our analyses, and is obtained by elementary trigonometric identities.

Lemma 27. Let $\zeta = \exp(i\theta)$ with $\theta = \frac{2k\pi}{m}$ and $\gcd(k, m) = 1$ be a m-th primitive root of the unity, and $d = \phi(m)$. Let $S(\theta) = \sum_{j=0}^{d-1} \zeta^{2j}$. We have $S(\theta) = \frac{\sin(\theta d)}{\sin\theta} e^{i\theta(d-1)}$ and

$$\operatorname{Re} S(\theta) = \frac{1}{2} + \frac{\sin((2d-1)\theta)}{2\sin\theta} \quad \text{and} \quad \operatorname{Im} S(\theta) = \frac{\sin(d\theta)\sin((d-1)\theta)}{\sin\theta}.$$

Remark. If m is a power of 2 then $2d = m$ so we always have $S(\theta) = 0$.

4.2.2 $\mathscr{K}_\mathbb{R}$-valued matrices

For $Q \in \mathscr{K}_\mathbb{R}^{2\times 2}$, denote Q^* as its conjugate-transpose, where $*$ is the conjugation in $\mathscr{K}_\mathbb{R}$. Positive definiteness extends to such matrices: we say Q is *totally positive definite* when $Q = Q^*$ and all the d matrices $\varphi_i(Q)$ induced by the field embeddings are hermitian positive definite. We then write $Q \succ 0$. We define a $\mathscr{K}_\mathbb{R}$-bilinear form $\langle \mathbf{x}, \mathbf{y} \rangle_Q = \mathbf{x}^* Q \mathbf{y}$. With the canonical embedding, we define an euclidean norm on $\mathcal{H}$ as $\|\varphi(\mathbf{x})\|_Q^2 = \sum_i \varphi_i(\langle \mathbf{x}, \mathbf{x} \rangle_Q)$.

With the defined norm and $\mathscr{K}_{\mathbb{R}}$-bilinear form, the Gram-Schmidt orthogonalization procedure for a pair of linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathscr{K}^2$ is defined as

$$\widetilde{\mathbf{b}}_1 := \mathbf{b}_1, \widetilde{\mathbf{b}}_2 := \mathbf{b}_2 - \frac{\langle \mathbf{b}_1, \mathbf{b}_2 \rangle_Q}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle_Q} \cdot \widetilde{\mathbf{b}}_1.$$

It is easy to check that $\langle \widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2 \rangle_Q = 0$. The Gram-Schmidt matrix with columns $\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2$ is denoted by $\widetilde{\mathbf{B}}$ and according to standard linear algebra computation, we have $\det \widetilde{\mathbf{B}} = \det \mathbf{B}$. For a given form Q, we let $|\mathbf{B}|_{\mathscr{K},Q} = \max(||\varphi(\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle)_Q||_\infty, ||\varphi(\langle \widetilde{\mathbf{b}_2}, \widetilde{\mathbf{b}_2} \rangle)_Q||_\infty)^{1/2}$. When there is no subscript Q, it is implied that $Q = \mathbf{I}_2$.

4.2.3 NTRU lattices

In this work, we only consider a free $\mathscr{R}$-modules of rank 2 in $\mathscr{K}^2$. Suppose that this rank 2 free module has a basis $(\mathbf{x}, \mathbf{y})$. Then the free module is all the $\mathscr{R}$-linear combinations of the basis $(\mathbf{x}, \mathbf{y})$. In other words, the free module is $\mathscr{M} = \mathscr{R}\mathbf{x} + \mathscr{R}\mathbf{y}$ where $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$. We define $\mathscr{K}_{\mathbb{R}}^2$ with a totally positive definite form Q and its corresponding inner product. Suppose that $\mathbf{B}$ is the basis matrix for $\mathscr{M}$, the volume of the associated lattice is $\mathrm{vol}_Q(\mathscr{M}) = \sqrt{N_{\mathscr{K}}\big(\det \mathbf{B}^* Q \mathbf{B}\big)}$.

Given $f, g \in \mathscr{R}$ such that f is invertible modulo some prime $q \in \mathbb{Z}$, we let $h = f^{-1}g \mod q$. The NTRU module determined by h is $\mathscr{L}_{\text{NTRU}} = \{(u, v) \in \mathscr{R}^2 : uh - v = 0 \mod q\}$. Two bases of this free module are often used for our purpose:

$$\mathbf{B}_h = \begin{bmatrix} 1 & 0 \\ h & q \end{bmatrix} \quad \text{and} \quad \mathbf{B}_{f,g} = \begin{bmatrix} f & F \\ g & G \end{bmatrix},$$

where $F, G \in \mathscr{R}$ are such that $fG - gF = q$ and (F, G) should have relatively small norm. This free module can be regarded as a lattice of volume $q^d \sqrt{N_{\mathscr{K}}(\det Q)}$ in $(\mathbb{R}^{2d}, Q)$ in the coefficient embedding.

Lemma 28. [EFG+22] Let $\mathbf{B}_{f,g}$ be a basis of an NTRU module and $\mathbf{b}_1 = (f, g)$. We have $\sqrt{q} N_{\mathscr{K}}(\det Q)^{1/(4d)} \leq |\mathbf{B}_{f,g}|_{\mathscr{K},Q}$ and

$$|\mathbf{B}_{f,g}|_{\mathscr{K},Q}^2 = \max\left(||\varphi(\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle_Q)||_\infty, \left\| \frac{q^2 \cdot \det Q}{\varphi(\langle \widetilde{\mathbf{b}_1}, \widetilde{\mathbf{b}_1} \rangle_Q)} \right\|_\infty \right)$$

4.3 New trapdoor algorithms for hybrid sampling

4.3.1 NTRU trapdoors in Falcon and Mitaka

In this section, for the sake of simplicity, we explain the trapdoor generation algorithm in the power-of-two cyclotomic case, and with the $\mathscr{K}_{\mathbb{R}}$-bilinear form associated to $Q = \mathbf{I}_2$.

With respect to Prest's hybrid sampler, an NTRU trapdoor $\mathbf{B}_{f,g}$ has a quality α defined as

$$\alpha = |\mathbf{B}_{f,g}|_{\mathscr{K}}/\sqrt{q}, \tag{4.1}$$

where we recall that $|\mathbf{B}_{f,g}|^2_{\mathscr{K}} = \max\left(\|\varphi(ff^* + gg^*)\|_\infty, \left\|\frac{q^2}{\varphi(ff^*+gg*)}\right\|_\infty\right)$. Quality with respect to the Klein–GPV sampler admits a similar expression.

In hash-and-sign signatures, security againt forgery attacks is driven by the standard deviation of the sampler, which is essentially $\alpha\sqrt{q}$. As the smaller the value of α, the harder forgery becomes. The goal of key generation in schemes such as DLP [DLP14], Falcon [PFH+22] and Mitaka [EFG+22] is to construct in reasonable time bases $\mathbf{B}_{f,g}$ with α as small as possible (and in particular, smaller than a given threshold related to the acceptance radius of signature verification).

An important observation regarding NTRU trapdoors is that the knowledge of the first basis vector (f, g) alone is sufficient to determine the quality of the whole basis (see for example Lemma 28 for Mitaka). As a result, to test if a vector (f, g) can be completed into a trapdoor $\mathbf{B}_{f,g}$ reaching the desired quality threshold, it is not necessary to compute the second vector (F, G), which is a notoriously costly operation, even accounting for optimizations such as [PP19].

In DLP, Falcon and Mitaka, trapdoors are then generated by trial-and-error, by generating many potential candidate first vectors (f, g) and testing whether they satisfy the required quality threshold. The candidates themselves are generated as discrete Gaussian vectors in $\mathscr{R}^2$ with the correct expected length. In that way, Falcon reaches quality $\alpha = 1.17$ with respect to its FFO-based sampler (that admits the same quality metric as Klein–GPV). Doing this directly for the hybrid sampler, as discussed in [Pre15], only achieves quality $\gtrsim 3$ in dimension 512, and even larger in higher dimensions. As a result, the Mitaka paper has to introduce randomness recycling and other techniques on top of this general approach in order to increase the number of candidates and improve achievable quality; with those improvements, Mitaka acheives quality $\alpha = 2.04$ in dimension 512 (which translates to 20 fewer bits of security compared to Falcon, and is thus unfortunately not sufficient to reach NIST security level I).

4.3.2 Antrag: annular NTRU trapdoor generation

The main contribution of this work is a novel NTRU trapdoor generation algorithm for the hybrid sampler, which achieves much better quality than Mitaka, and reaches the same security NIST levels as Falcon.

The intuition behind our new approach stems from the following observation. For a fixed $\alpha \geq 1$, requiring a trapdoor $\mathbf{B}_{f,g}$ to satisfy $|\mathbf{B}_{f,g}|_{\mathscr{K}} \leq \alpha\sqrt{q}$ is equivalent to

enforcing that for all $1 \leq i \leq d$, we have

$$\frac{q}{\alpha^2} \leq |\varphi_i(f)|^2 + |\varphi_i(g)|^2 \leq \alpha^2 q, \tag{4.2}$$

(where we recall that the $\varphi_i(f)$ are the *embeddings* of f in $\mathbb{C}$, and similarly for g). Equivalently, this means that for all i, the pair $\big(|\varphi_i(f)|, |\varphi_i(g)|\big)$ belongs to the arc of annulus $A^+_\alpha := A^+\big(\sqrt{q}/\alpha, \alpha\sqrt{q}\big)$.

It is thus natural to try and sample f and g *from their embeddings* (i.e., in the Fourier domain), by picking the pairs $\big(\varphi_i(f), \varphi_i(g)\big)$ as *uniform* random pairs of complex numbers such that satisfying the condition that the pair of their magnitudes belongs to A^+_α: in other words, pick (x_i, y_i) uniformly at random in A^+_α and then sample $\varphi_i(f)$ and $\varphi_i(g)$ as uniform complex numbers of magnitures x_i and y_i respectively. Note that only $d/2$ pairs are needed, as the remaining ones are determined by conjugation.

Moreover, sampling uniformly in an annulus (or, as in our case, an arc of annulus) in polar coordinates (ρ, θ) is easy: it suffices to sample the angle θ and the *square* ρ^2 of the radial coordinate uniformly in their respective ranges. This is because the area element in polar coordinates in $\rho\, d\rho\, d\theta = \frac{1}{2} d(\rho^2)\, d\theta$. This gives rise to Algorithm 1 for the sampling on the pairs of embeddings.

However, one soon realizes that the real polynomials $\tilde{f}, \tilde{g}$ corresponding to the embeddings generated by the Algorithm 1 (via the inverse Fourier transform φ^{-1}) do not always have integer coefficients, and hence do not generally correspond to ring elements. In general, they are arbitrary elements of the $\mathbb{R}$-algebra $\mathscr{K}_\mathbb{R}$.

In order to obtain actual ring elements, a natural solution is to round those real polynomials $\tilde{f}, \tilde{g}$ coefficient-wise. This yields $f = \lfloor \tilde{f} \rceil$ and $g = \lfloor \tilde{g} \rceil$ in $\mathscr{R}$, which are potential candidates for a trapdoor. It turns out, however, that if one starts from $\tilde{f}, \tilde{g}$ uniform with their embeddings of magnitude in A^+_α, the resulting rounded ring elements are very unlikely to also have their embeddings of magnitude in that arc of annulus. Thus, they do not typically give rise to a trapdoor of the desired qual-

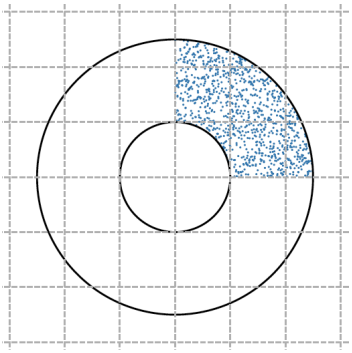


Figure 4.1: $(|z|, |w|)$ is sampled uniformly in the annulus $A^+(r, R)$.

Input: $0 < r < R$, the radii of $A^+(r, R)$
Result: $z, w \in \mathbb{C}$ uniformly such that $\big(|z|, |w|\big) \in A^+(r, R)$
$u \hookleftarrow \mathcal{U}\left([r^2, R^2]\right)$;
$\rho \leftarrow \sqrt{u}$;
$\theta \hookleftarrow \mathcal{U}\left([0, \pi/2]\right)$;
$(x, y) \leftarrow \big(\rho \cos\theta, \rho \sin\theta\big)$; /* $(\rho_a, \rho_b) \leftarrow \mathcal{U}(A(r, R))$ */
$\omega, \omega' \hookleftarrow \mathcal{U}\left([0, 2\pi]\right)$;
$(z, w) \leftarrow \big(x \cdot e^{i\omega}, y \cdot e^{i\omega}\big)$;
return (z, w);

Algorithm 1: Candidate pairs from uniform annulus sampling

ity. This is because rounding adds an additive term (essentially uniformly distributed in $[-1/2, 1/2)$) to each coefficient, which translates to an additive "error" on each embedding, making it unlikely that the embeddings all remain in the desired domain.

A straightforward workaround is to compensate this decoding error by sampling the embeddings of $\tilde{f}, \tilde{g}$ with their magnitude in a narrower annulus $A^+\big((1/\alpha+\varepsilon)\sqrt{q}, (\alpha-\varepsilon)\sqrt{q}\big)$. This yields Algorithm 2, which is our proposed Antrag trapdoor generation algorithm.

Remark. One could consider carrying out the decoding to the ring differently, for example by sampling discrete Gaussians f and g in $\mathscr{R}$ centered at $\tilde{f}$ and $\tilde{g}$ respectively. The resulting algorithm would be simpler to analyze in some ways, and might be seen as better behaved in a certain sense, but it does have a major drawback: it introduces a much larger decoding error (on the order of the smoothing parameter $\eta_\epsilon(\mathbb{Z})$ of $\mathbb{Z}$ on each coefficient, instead of the standard deviation $1/\sqrt{12}$ of the uniform distribution in $[-1/2, 1/2)$, so about 4 times larger). As a result, in this work, we focus on the rounding approach.

Since the magnitude of the rounding error is indepedent of q, however, the Gaussian decoding approach could be preferred in settings where q is chosen larger than in schemes like Falcon and Mitaka (e.g., identity-based encryption and other more advanced applications of GPV-style trapdoors).

4.3.3 Error analysis

We have mentioned above that taking the magnitudes of the embeddings of $\tilde{f}$ and $\tilde{g}$ in A^+_α was very unlikely to result in f and g of the required quality α after rounding, but that the probability increased greatly when choosing $\tilde{f}$ and $\tilde{g}$ with embedding magnitudes in a narrower arc of annulus $A^+(r, R)$ with

$$r = \big(1/\alpha + \varepsilon\big)\sqrt{q} \quad \text{and} \quad R = \big(\alpha - \varepsilon\big)\sqrt{q}.$$

Input: The degree d, a target quality α, a correction parameter ε, and q
Result: $f, g \in \mathscr{R}^2$ such that $\frac{q}{\alpha^2} \leq |\varphi_i(f)|^2 + |\varphi_i(g)|^2 \leq \alpha^2 q$ for $\forall i$.
$(r, R) \leftarrow \big((1/\alpha + \varepsilon)\sqrt{q}, (\alpha - \varepsilon)\sqrt{q}\big)$;
repeat
  for $1 \leq i \leq d/2$ **do**
    using Algorithm 1, sample $(z_i, w_i) \in \mathbb{C}^2$ uniformly such that $\big(|z_i|, |w_i|\big) \in A^+(r, R)$.
  end
  $\tilde{f} \leftarrow \varphi^{-1}(z_1, \ldots, z_{d/2}) \in \mathscr{K}_{\mathbb{R}}$;
  $\tilde{g} \leftarrow \varphi^{-1}(w_1, \ldots, w_{d/2}) \in \mathscr{K}_{\mathbb{R}}$;
  $f \leftarrow \lfloor \tilde{f} \rceil$;
  $g \leftarrow \lfloor \tilde{g} \rceil$;
until $\big(|\varphi_i(f)|, |\varphi_i(g)|\big) \in A^+\big(\sqrt{q}/\alpha, \alpha\sqrt{q}\big)$ *for all* $i = 1, \ldots, d/2$;
return (f, g)

Algorithm 2: Antrag trapdoor generation

In this section, we would like to quantify this claim, based both on a heuristic analysis of the success probability, and on simulation data. Concretely, write $e = (e_f, e_g) = \big(f - \tilde{f}, g - \tilde{g}\big) \in \mathscr{K}_{\mathbb{R}}^2$ for the error term introduced by rounding. We would like to control the distribution of the embeddings of e_f and e_g in order to estimate the likelihood that the condition $\big(|\varphi_i(f)|, |\varphi_i(g)|\big)$ will be satisfied for all i.

In the polynomial basis, we write:

$$e_f = \sum_{j=0}^{d-1} e_f^{(j)} x^j$$

and similarly for e_g. Heuristically, we expect the coefficients $e_f^{(j)}$ and $e_g^{(j)}$ to behave essentially like independent uniform random variables in $[-1/2, 1/2)$.[4] This is well-supported by experiments (see Figure A.1a in Supplementary Material A).

Now consider a single embedding φ_0, and recall that we are interested in an a priori arbitrary cyclotomic base ring, so that φ_0 is defined by the evaluation at some primitive m-th root of unity $\zeta = e^{i\theta}$. We therefore have:

$$\varphi_0(e_f) = x_0 + iy_0 \quad \text{with} \quad x_0 = \sum_{j=0}^{d-1} e_f^{(j)} \cos(j\theta) \quad \text{and} \quad y_0 = \sum_{j=0}^{d-1} e_f^{(j)} \sin(j\theta).$$

This expresses the real and imaginary parts x_0, y_0 of $\varphi_0(e_f)$ as the sum of d independent random variables, with d relatively large, so by the central limit theorem, $\varphi_0(e_f)$

[4] This is equivalent to saying that the distribution of $\tilde{f}$ and $\tilde{g}$ is uniform modulo $\mathscr{R}$ in $\mathscr{K}_{\mathbb{R}}$, which should indeed happen as soon as we have sufficient width (i.e., if we exceed a regularity metric analogous to the smoothing parameters for Gaussians).

should essentially behave[5] like a normal random variable in $\mathbb{C}$, essentially determined by its expectation and covariance.

Now since $e_f^{(j)}$ has mean 0 and variance $1/12$ for all j, we obtain that $\mathbb{E}[x_0] = \mathbb{E}[y_0] = 0$. Therefore, the pair (x_0, y_0) has mean 0, and its covariance matrix is easily expressed as follows:

$$\begin{aligned}\Sigma &= \sum_{j=0}^{d-1} \mathrm{Var}[e_f^{(j)}] \cdot \begin{bmatrix} \cos^2(j\theta) & \cos(j\theta)\sin(j\theta) \\ \cos(j\theta)\sin(j\theta) & \sin^2(j\theta) \end{bmatrix} \\ &= \frac{1}{12}\sum_{j=0}^{d-1} \frac{1}{2}\begin{bmatrix} 1+\cos(2j\theta) & \sin(2j\theta) \\ \sin(2j\theta) & 1-\cos(2j\theta) \end{bmatrix} = \frac{d}{24}\mathbf{I}_2 + E(\theta),\end{aligned}$$

where

$$E(\theta) = \frac{1}{12}\begin{bmatrix} \mathrm{Re}\, S(\theta) & \mathrm{Im}\, S(\theta) \\ \mathrm{Im}\, S(\theta) & -\mathrm{Re}\, S(\theta) \end{bmatrix}.$$

Thus, we expect that $\varphi_0(e_f)$ follows the normal distribution $\mathcal{N}(0, \Sigma)$, and the same argument applies to $\varphi(e_g)$ as well. Moreover, heuristically, those two normal distributions should be independent (this is again well-verified in practice: see Figure A.1b).

At this point, we would therefore like to estimate the probability that the rounded pair (f, g) satisfies the quality condition at embedding φ_0, i.e., that the following inequality is satisfied:

$$q/\alpha^2 \leq |\varphi_0(f)|^2 + |\varphi_0(g)|^2 \leq \alpha^2 q.$$

Now, the quantity $|\varphi_0(f)|^2 + |\varphi_0(g)|^2$ is just the squared Euclidean (or Hermitian) norm $\|\mathbf{v}\|^2$ of $\mathbf{v} := \varphi_0\big((f, g)\big)$ in $\mathbb{C}^2$. If we also write $\widetilde{\mathbf{v}} := \varphi_0\big((\tilde{f}, \tilde{g})\big)$ and $\mathbf{e} := \varphi_0\big((e_f, e_g)\big)$, we have $\mathbf{v} = \widetilde{\mathbf{v}} + \mathbf{e}$, and therefore:

$$\|\mathbf{v}\|^2 = \|\widetilde{\mathbf{v}}\|^2 + \|\mathbf{e}\|^2 + 2\cos(\nu)\|\widetilde{\mathbf{v}}\| \cdot \|\mathbf{e}\|$$

where ν the angle between the vectors $\widetilde{\mathbf{v}}$ and $\mathbf{e}$.

Write $X = \|\widetilde{\mathbf{v}}\|^2$, $Y = \|\mathbf{e}\|^2$ and $Z = \|\mathbf{v}\|^2$, so that:

$$Z = X + Y + 2\sqrt{X}\sqrt{Y}\cos\nu.$$

We have a good heuristic understanding of how these random variables behave.

X is completely controlled: by the annular sampling algorithm, it is uniform in $[r^2, R^2]$.

Y was described by the previous discussion: it is the sum $|\varphi_0(e_f)|^2 + |\varphi_0(e_g)|^2$, so the sum of the squared Euclidean norms of two normal random variable $\mathcal{N}(0, \Sigma)$.

[5] This can in fact be made rigorous with Berry-Esseen's inequality.

Therefore, it is the sum of a $\chi^2(2)$ random variable scaled by the first eigenvalue of Σ and a $\chi^2(2)$ scaled by the second eigenvalue of Σ.

And finally, since the distribution of $(\tilde{f}, \tilde{g})$ is isotropic and a priori independent from the rounding term $\mathbf{e}$ (again by a heuristic regularity assumption), the angle ν between the vectors $\widetilde{\mathbf{v}}$ and $\mathbf{e}$ should be uniform in $[0, 2\pi)$, and the three variables X, Y and ν should be essentially independent.

This lets us completely estimate the desired probability of success for embedding φ_0, namely $\mathbb{P}\big[q/\alpha^2 \le Z \le \alpha^2 q\big]$.

Power-of-two cyclotomic fields.

In particular, in the case of power-of-two cyclotomic fields, the situation is made comparatively simple by the fact that $E(\theta) = 0$ for all embeddings as guaranteed by Lemma 27, and hence the covariance matrix Σ is just $\frac{d}{24}\mathbf{I}_2$.

This means that the variable Y simply follows the χ^2 distribution with 4 degrees of liberty scaled by $d/24$, i.e., $Y \sim \frac{d}{24}\chi^2(4)$. This distribution has a particularly simple CDF, characterized by the formula:

$$\mathbb{P}\Big[Y > t \cdot \frac{d}{24}\Big] = \Big(1 + \frac{t}{2}\Big)\exp(-t/2) \quad \text{for all } t \ge 0. \tag{4.3}$$

Now recall that we want to estimate the probability of success $p_{\text{succ}} := \mathbb{P}\big[q/\alpha^2 \le Z \le \alpha^2 q\big]$. Clearly, we have

$$p_{\text{succ}} = 1 - \mathbb{P}\big[Z > \alpha^2 q\big] - \mathbb{P}\big[Z < \frac{q}{\alpha^2}\big]. \tag{4.4}$$

We compute the probability $\mathbb{P}\big[Z > \alpha^2 q\big]$ and $\mathbb{P}\big[Z < q/\alpha^2\big]$ seperately.

On the one hand, the inequality $Z > \alpha^2 q$, is equivalent to:

$$X + Y + 2\sqrt{X}\sqrt{Y}\cos\nu - \alpha^2 q > 0. \tag{4.5}$$

If we consider the left-hand side as a quadratic trinomial in the variable $\sqrt{Y}$, it has discriminant $\Delta_1 = 4\alpha^2 q - 4X\sin^2\nu$. Since $X < \alpha^2 q$ is guaranteed by our sampling algorithm, this discriminant always satisfies $\Delta_1 > 0$. By Vieta's formula, the product of the two roots is $X - \alpha^2 q < 0$, which means that one root is positive and the other one is negative. Thus, $Z > \alpha^2 q$ if and only if $\sqrt{Y}$ is greater than the positive root of the trinomial:

$$Z > \alpha^2 q \quad \text{if and only if} \quad \sqrt{Y} > -\sqrt{X}\cos\nu + \sqrt{\Delta_1}. \tag{4.6}$$

On the other hand, the inequality $Z < q/\alpha^2$ is equivalent to:

$$X + Y + 2\sqrt{X}\sqrt{Y}\cos\nu - \frac{q}{\alpha^2} < 0. \tag{4.7}$$

Again, regarding this as a quadratic inequality in $\sqrt{Y}$, the discriminant of the trinomial becomes $\Delta_2 = 4q/\alpha^2 - 4X\sin^2\nu$. If $\Delta_2 \le 0$, inequality (4.7) can never be satisfied as the trinomial has no real roots. If $\Delta_2 > 0$, according to Vieta formula, the product $X - q/\alpha^2$ of the two roots is always positive (so the two roots are either both positive or both negative), because $X > \frac{q}{\alpha^2}$ is guaranteed by our sampling algorithm. Since $\sqrt{Y}$ is non-negative, inequality (4.7) can only happen when both roots are positive, or equivalently when their sum $-2\sqrt{X}\cos\nu$ is positive (which is equivalent to $\cos\nu < 0$, i.e., $\nu \in (\pi/2, 3\pi/2)$). If that condition is satisfied, the inequality is equivalent to $\sqrt{Y}$ being between the two roots of the trinomial. Therefore:

$$Z < \frac{q}{\alpha^2} \quad \text{if and only if} \quad \Delta_2 > 0 \text{ and } \nu \in (\pi/2, 3\pi/2) \text{ and} \\ -\sqrt{X}\cos\nu - \sqrt{\Delta_2} < \sqrt{Y} < -\sqrt{X}\cos\nu + \sqrt{\Delta_2}. \tag{4.8}$$

Now, define the function $P(t)$ for $t \ge 0$ as follows:

$$P(t) := \Big(1 + \frac{t^2}{2\sigma^2}\Big)\exp(-t^2/2\sigma^2) \quad \text{where} \quad \sigma = \sqrt{\frac{d}{24}}. \tag{4.9}$$

By property (4.10) of the $\chi^2(4)$ distribution of Y, we see that, for all $t \ge 0$:

$$\mathbb{P}\Big[\sqrt{Y} > t\Big] = \mathbb{P}\Big[Y > t^2\Big] = \Big(1 + \frac{t^2/\sigma^2}{2}\Big)\exp(-t^2/2\sigma^2) = P(t).$$

We also fix the following notation:

$$\begin{aligned}
\beta(X,\nu) &:= -\sqrt{X}\cos\nu + \sqrt{\alpha^2 q - X\sin^2\nu}, \\
\gamma_1(X,\nu) &:= -\sqrt{X}\cos\nu + \sqrt{\frac{q}{\alpha^2} - X\sin^2\nu}, \\
\gamma_2(X,\nu) &:= -\sqrt{X}\cos\nu - \sqrt{\frac{q}{\alpha^2} - X\sin^2\nu}.
\end{aligned}$$

Then, in view of (4.6), we have:

$$\mathbb{P}\big[Z > \alpha^2 q\big] = \mathbb{E}\big[P\big(\beta(X,\nu)\big)\big] = \frac{1}{2\pi(R^2 - r^2)}\int_0^{2\pi}\int_{r^2}^{R^2} P\big(\beta(X,\nu)\big)\,dX\,d\nu.$$

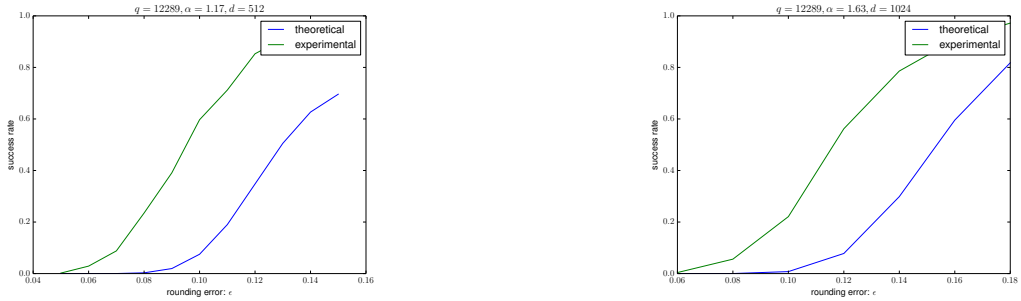
Similarly, in view of (4.8), we have:

$$\begin{aligned}
\mathbb{P}\big[Z < q/\alpha^2\big] &= \mathbb{E}\Big[[\![\nu \in (\pi/2, 3\pi/2) \text{ and } q/\alpha^2 > X\sin^2\nu]\!] \cdot \Big(P\big(\gamma_2(X,\nu)\big) - P\big(\gamma_1(X,\nu)\big)\Big)\Big] \\
&= \frac{1}{2\pi(R^2 - r^2)}\int_{\pi/2}^{3\pi/2}\int_{r^2}^{R^2} [\![q/\alpha^2 > X\sin^2\nu]\!] \cdot \Big(P\big(\gamma_2(X,\nu)\big) - P\big(\gamma_1(X,\nu)\big)\Big)\,dX\,d\nu.
\end{aligned}$$

where $[\![C]\!]$ is the Iverson bracket notation (evaluating to 1 if condition C is true, and 0 otherwise).

The last two formulas, combined with (4.4), give us an expression of the probability of success p_{succ} on embedding φ_0 in terms of the double integral of well-behave functions on a simple domain of integration. This is very easy to evaluate numerically, and we verify that the results very closely follow simulations on a given embedding.

Once the job is done for a single embedding, we are then tempted to estimate the probability of success for *all* $d/2$ embeddings as simply $p_{\text{succ}}^{d/2}$, assuming heuristically that the rounding errors on all the embeddings behave independently of each other.



(a) $\alpha = 1.17, d = 512, q = 12289$ (b) $\alpha = 1.63, d = 1024, q = 12289$

Figure 4.2: The probability that f, g satisfies requirement after rounding w.r.t the rounding error: theoretical vs. experimental

This is, unfortunately, not a completely plausible model of what happens in practice: indeed, since the vector (e_f, e_g) is uniform in a hypercube, its image under the map φ (which, in our case, is orthogonal) is still uniform in a hypercube (just not with axes parallel to the basis vectors of the canonical basis), and therefore it is not accurately modeled as a joint spherical Gaussian distribution. Due to the fact that the uniform distribution has thin tails (it is platykurtic), the joint distribution is less likely to be large and hence cause (f, g) to end up outside of the annulus A_α^+ compared to the independent case.

This is indeed what we observe in experiments, as shown in Figure 4.2, where we compare the theoretical values obtained from the integrals above with the experimental success rate of Algorithm 2.

In any case, even with the pessimistic model of independent embeddings of the error vector, and with parameters $q = 12289$, $d = 512$ and $\alpha = 1.17$ (the same as the parameters of Falcon–512), we would get a good success probability of around 20% for the whole vector when taking a margin factor of $\varepsilon = 0.11$, which is far from causing any security issue, as will be shown in the next section. In practice, however,

we achieve an even better success rate with $\varepsilon = 0.08$, which is the value we pick in our parameter selection.

General cyclotomic fields.

Over non power-of-two cyclotomics, the previous discussion carries over for the most part, except for one change: in the expression of the covariance matrix Σ of (x_0, y_0), namely:

$$\Sigma = \frac{d}{24}\mathbf{I}_2 + E(\theta),$$

the matrix $E(\theta)$ no longer vanishes in general, and thus Σ usually has distinct eigenvalues $\sigma_1^2 = \frac{1}{24}(d + |S(\theta)|)$ and $\sigma_2^2 = \frac{1}{24}(d - |S(\theta)|)$.

It follows, as discussed previously, that the random variable $Y = \|\mathbf{e}\|^2$ is now distributed like the sum of a $\chi^2(2)$ distribution scaled by σ_1^2 and an independent $\chi^2(2)$ distribution scaled by σ_2^2. This is also known as the $\mathrm{GDC}(1, \frac{1}{2\sigma_1^2}; 1, \frac{1}{2\sigma_2^2})$ distribution (see [WWW+16, Equation 5]): the convolution of two Gamma distribution of suitable parameters. Fortunately, in this particular case, the distribution has a relatively simple CDF again. We actually have:

$$\mathbb{P}\Big[Y > t\Big] = \frac{\sigma_1^2 \exp(-t/2\sigma_2^2) - \sigma_2^2 \exp(-t/2\sigma_1^2)}{\sigma_1^2 - \sigma_2^2} \quad \text{for all } t \geq 0. \tag{4.10}$$

Therefore, we can directly apply the previous discussion by simply replacing the function $P(t)$ of (4.9) by the following one in this setting:

$$P(t) := \frac{\sigma_1^2 \exp(-t^2/2\sigma_2^2) - \sigma_2^2 \exp(-t^2/2\sigma_1^2)}{\sigma_1^2 - \sigma_2^2},$$

and all the formulas for p_{succ} then carry over. One just has to take into account that the value of p_{succ} now depends on θ and hence on the embedding, so if we want to apply the pessimistic heuristic that the success probabilities on distinct embeddings are independent, we have to multiple all the a priori distinct values together.

Qualitatively speaking, the behavior in this case is in fact quite close to the power-of-two case, since for most embeddings, $|S(\theta)| = \left|\frac{\sin(d\theta)}{\sin\theta}\right|$ is small compared to d: $\sin\theta$ is bounded away from zero except possibly for just a handful of embeddings with θ close to a multiple of π.

For those few embeddings, success probability tends to become slightly worse due to the longer tail of Y. Nevertheless, even in the worst case, which is the first embedding of a cyclotomic field of conductor $m = 2^\ell 3^k$ (hence $d = m/3$), we have:

$$S(\theta) = \frac{\sin(2\pi/3)}{\sin(2\pi/3d)} \approx \frac{\sqrt{3}/2}{2\pi/3d} = \frac{3\sqrt{3}}{4\pi}d \approx 0.413d$$

, so the standard deviation of the error in the longest direction is increased at most by a factor of $\approx \sqrt{1.413} < 1.2$.

4.4 Security analysis

Security of signatures is considered with regards to two notions: on the one hand, *security against forgery*—namely the ability for an attacker to forge a valid signature with only a message and the public key—and *security against key recovery*—i.e. fully recovering the secret key with only the datum of the public key.

Since our new algorithms only change the set of secret keys in which trapdoors are found, there is no new forge attacks stemming from this modification— only the security level has to be reevaluated, favourably for us as the quality of the trapdoors are improved.

Hence, this section deals with the impact of the new key generation on the resilience of the scheme only against *key recovery*. We first recall the state-of-the-art attacks on such NTRU lattices and then examine precisely the resilience against so-called subfield type attacks, which might be relevant for the new parameters. In the following, we compute expectations on the norm of f, g and related quantities as if they were drawn under continuous distributions (and not discrete). This eases the presentation of the results and could be make formal using standard subgaussians arguments. In particular, we heuristically assume that all pair of embeddings $(\varphi_i(f), \varphi_i(g))$ of a secret key (f, g) sampled from Algorithm 2 are distributed uniformly such that their magnitudes are in $A^+(\sqrt{q}/\alpha, \alpha\sqrt{q})$.

4.4.1 Classical attack against NTRU keys

The key recovery amounts to the problem of finding a private secret key with small norm (i.e. $(f, g) \in \mathscr{R}^2$) with the knowledge of the public available elements q and h. For the regime of parameters in Falcon or Mitaka, to the best of our knowledge, the known best attacks are realized through lattice reduction. It works as the following steps: first construct the algebraic lattice over $\mathscr{R}$ spanned by the vectors $(0, q)$ and $(1, h)$. Then look for the lattice vector $s = (f, g)$ among all possible lattice vectors of norm bounded by $\|s\|$ (or a functionally equivalent vector, for instance $(\mu \cdot f, \mu \cdot g)$ for any unit μ of the ring of integer of the number field).

Recall that by construction and our modelization, the expected (squared) norm of $\|s\|$ concentrates at qA for $A = \frac{1}{2}(\alpha^2 + \alpha^{-2})$[6]. We make use of the so-called *projection trick* to avoid enumerating and testing over all the sphere of radius $\sqrt{q}S$ (which contains around $\left(\frac{qA}{q}\right)^d = A^d$ vectors under the Gaussian heuristic[7]). More

[6]Each pair of embedding $(\varphi(f), \varphi(g))$ satisfies that $|\varphi(f)|^2 + |\varphi(g)|^2$ is uniform in the interval $[q\alpha^{-2}, q\alpha^2]$, we then sum over all embeddings to retrieve $\|f\|^2 + \|g\|^2$.

[7]The Gaussian heuristic predicts the number of vectors of length at most ℓ in a random lattice Γ of volume V to be a $v_\Gamma(\ell)/V + o(1)$ for large enough ℓ, where $v_\Gamma(\ell)$ is the volume of the sphere of radius ℓ for the measure induced by the inner product on Γ.

sepecifically, we do the following things [ETWY22]: first denote β as the block size for the DBKZ algorithm [MW16] and start by reducing the public basis with this DBKZ algorithm. If $[\mathbf{b}_1, \ldots, \mathbf{b}_{2d}]$ is the output of DBKZ. Then if we can recover the *projection* of the secret key onto $\mathcal{P}$, the orthogonal space to $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{2d-\beta-1})$, then we can recover in polynomial time the full key by *Babai nearest plane* algorithm to lift it to a lattice vector of the desired norm. Therefore, for our purpose, it is enough to find the projection of the secret key among the shortest vector of the lattice generated by the last β vectors projected onto $\mathcal{P}$. Classically, sieving on this projected lattice will recover all vectors of norm smaller than $\sqrt{\frac{4}{3}}\ell$, where ℓ is the norm of the $2d-\beta$-th Gram-Schmidt vector $\widetilde{\mathbf{b}}_{2d-\beta}$ of the reduced basis.

The expected length of the projection is usually estimated under the *Geometric Series Assumption* (GSA)[8]. Instantiated on NTRU lattices, it states that the Gram-Schmidt vectors of the basis outputted by DBKZ with block-size β satisfy the relations (see Cor 2. of [MW16]):

$$\|\widetilde{\mathbf{b}}_i\| = \delta_\beta^{2(d-i)+1}\sqrt{q} \quad \text{where} \quad \delta_\beta = \left(\frac{(\pi\beta)^{1/\beta}\cdot\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}.$$

Therefore, we expect that

$$\ell = \delta_\beta^{-2(d-\beta)+1}\sqrt{q} \approx \sqrt{q}\cdot\left(\frac{\beta}{2\pi e}\right)^{1-\frac{d}{\beta-1}}.$$

Moreover, assuming that $\mathbf{s}$ behaves as a random vector, and using the GSA to bound the norm of the Gram-Schmidt vectors $[\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{2d-\beta}]$, the (squared) norm of its projection over $\mathcal{P}$ concentrates around

$$\frac{\beta}{2d}\cdot\mathbb{E}\left[\|\mathbf{s}\|^2\right] = \frac{Aq\beta}{2d}.$$

Hence, we will retrieve the projection among the sieved vectors if $\frac{Aq\beta}{2d} \leq \frac{4}{3}\ell^2$, that is if the following condition is fulfilled:

$$A \leq \frac{8d}{3\beta}\delta_\beta^{4(\beta-d)+2}. \tag{4.11}$$

4.4.2 Towards a subfield attack

Now suppose that the value of the (relative) norm $N = ff^* + gg^*$ is known exactly by the attacker[9]. Then it is possible to recover both summands ff^* and gg^* exactly.

[8]For practical estimation of the attacks, we can use numerical models coming from simulations instead of the GSA. In this section, we stick to the usual GSA model for the sake of simplicity and ease of exposition.

[9]This situation would happen for instance if the annulus was reduced to a circle, as in that case, N would simply be q^2.

Indeed, the vector (ff^*, gg^*) lives in the NTRU lattice of hh^* over the totally real subfield $\mathscr{K}^+$, that is to say that $ff^* \cdot hh^* = gg^* \pmod q$. Thus linear algebra reveals that $gg^* = \frac{Nhh^*}{1+hh^*} \bmod q$ and $fg^* = \frac{Nh^*}{1+hh^*} \bmod q$ over $\mathscr{R}$. As noted in [FKT+20], because f, g are chosen to be co-prime, the attacker then recovers a $\mathbb{Z}$-basis of of the principal ideal (g) on top of gg^* by a greatest common divisor computation between ideals. They finally retrieve g using the Gentry-Szydlo algorithm for power-of-two cyclotomic number fields or its extension for arbitrary cyclotomics (for instance appearing in [EFGT17]).

Now if the attacker does not know the value of N exactly, but has a fairly good approximation of it, the preliminary "linear algebra" part can be replaced by lattice reduction. Indeed, write $ff^* + gg^* = qN + E$ for a known N[10] and a small E and (ff^*, gg^*, E) is a small solution of the linear system :

$$\begin{cases} HX - Y = 0 \bmod q, \\ X + Y - E = qN, \end{cases} \tag{4.12}$$

where $H = hh^*$. Solving such a system amounts to finding a short vector inside the coset $(0, 0, qN) + \mathscr{L}$ (considered inside the extended NTRU lattice in $(\mathscr{K}^+)^3$ corresponding to $\{(u, v, w) | uH = v \pmod q\}$). A (row) basis of the lattice $\mathscr{L}$ corresponding to (4.12) is given by:

$$L = \begin{pmatrix} 1 & H & H+1 \\ 0 & q & q \end{pmatrix}.$$

and the most efficient known algorithms to solve this problem are essentially variations of lattice reduction and decoding (see for instance [EK20]), and amount in estimating the hardness of retrieving a vector of a given norm inside $\mathscr{L}$. We now give the details to find lower bound on the parameters of the keygen to make such attacks infeasible.

Remark. In this attack, we make two simplifying assumptions. We suppose that the attacker has access to the unrounded vector $(\tilde{f}, \tilde{g})$ (which we still identify to (f, g)), which is located in a smaller domain and hence a priori easier to attack. We also pretend that this vector actually consists of ring elements, so everything happens as if the attacker had access to more constrained ring elements than we are able to generate in practice. This is a stronger attacker model than reality, and hence a conservative way of assessing the power of this type of attacks in practice."

Distribution of the relative norm vector.

Let us first estimate the size of the error E we introduced above by describing the joint distribution of a pair of embeddings $(X_i, Y_i) = (\varphi_i(ff^*), \varphi_i(gg^*))$. Then, by

[10] The case where the term N is not exactly an integer but is close to be is treated in a similar manner by reducing to ISIS instead of SIS)

our modelization, we have $U = X_i + Y_i$ and $V = X_i - Y_i = U\cos(T)$ where U is a uniform variable over $[r^2, R^2]$ (for $r = \sqrt{q}(\alpha^{-1} + \epsilon), R = \sqrt{q}(\alpha - \epsilon)$ and T is uniform in $[0, \pi]$. Next we compute:

$$D := \text{Cov}\,(U, V) = \text{diag}\left(\frac{1}{12}\left(r^2 - R^2\right)^2, \frac{1}{6}\left(R^4 + R^2r^2 + r^4\right)\right).$$

Hence the covariance C of (X_i, Y_i) is equal to:

$$C = \frac{1}{4}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} D \begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{6}\begin{pmatrix}R^4 + r^4 & -\frac{1}{3}(R^4 - 4R^2r^2 + r^4)\\ -\frac{1}{3}(R^4 - 4R^2r^2 + r^4) & R^4 + r^4\end{pmatrix}$$

Recall that the canonical norm of $\mathscr{K}^+$ writes as $\|xx^*\|^2 = \frac{1}{d}\sum_i \varphi_i(xx^*)^2$. Treating the embeddings as an independent family of variables, and the middle-ring of the annulus for a convenient (public!) choice for N the expected norm of the error term E is

$$\mathbb{E}[\|E\|^2] = \mathbb{E}\left[\left\|ff^* + gg^* - \frac{1}{2}(R^2 + r^2)\right\|^2\right] = \frac{1}{12}\left(r^2 - R^2\right)^2 =: yq^2,$$

and in addition that

$$\mathbb{E}\left[\|ff^*\|^2 + \|gg^*\|^2\right] = \frac{d}{d}\text{tr}\,(C) = \frac{1}{16}(R^4 + r^4) =: 2xq^2.$$

Mounting the lattice attack.

By what precedes, we want to find a short solution of the system (4.12), where we know that $\|ff^*\|^2, \|gg^*\|^2 \approx xq^2$ and $\|E\|^2 = yq^2$ Notice the vector $(ff^*, gg^*, E) \in (\mathscr{K}^+)^3$ is therefore slightly unbalanced. Following the same rescaling technique as in Espitau et al. [ETWY22], we want to view the corresponding lattice problem under the twisted (Euclidean) norm encoded by the Gram matrix (of determinant 1)

$$G_\eta = \begin{pmatrix}\eta & 0 & 0\\ 0 & \eta & 0\\ 0 & 0 & \frac{1}{\eta^2}\end{pmatrix},$$

for $\eta = \left(\frac{y}{x}\right)^{\frac{1}{3}}$. Then under this new norm $\|\cdot\|_\eta$, we find that:

$$\begin{aligned}\mathbb{E}\left[\|(ff^*, gg^*, E)\|_\eta^2\right] &= \eta\mathbb{E}\left[\|ff^*\|^2\right] + \eta\mathbb{E}\left[\|gg^*\|^2\right] + \frac{\mathbb{E}\left[\|E\|^2\right]}{\eta^2}\\ &= 3q^2\left(x^2y\right)^{\frac{1}{3}}.\end{aligned}$$

Under this norm the lattice $\mathscr{L}$ has $\mathscr{K}^+$-volume:

$$\det(LG_\eta L^T) = \left| \begin{bmatrix} \eta H^2 + \eta + \frac{(H+1)^2}{\eta^2} & \eta Hq + \frac{(H+1)q}{\eta^2} \\ \eta Hq + \frac{(H+1)q}{\eta^2} & \eta q^2 + \frac{q^2}{\eta^2} \end{bmatrix} \right| = q^2 \left(\eta^2 + \frac{2}{\eta} \right),$$

giving a lattice of normalized volume being $\sqrt{q}(\eta^2 + \frac{2}{\eta})^{\frac{1}{4}}$ as of $\mathscr{K}^+$-rank 2. The attack is then similar as the one in section 4.4.1 but where we want to recover a vector of squared norm $3q^2(x^2y)^{\frac{1}{3}}$ in a $\mathbb{Z}$-lattice[11] of normalized (squared) volume $2q(\eta^2 + \frac{1}{\eta})^{\frac{1}{2}}$ of rank $2\frac{d}{2} = d$, yielding a condition of the form:

$$\frac{\beta}{d} 3q^2 \left(x^2 y\right)^{\frac{1}{3}} \leq 2q \left(\eta^2 + \frac{2}{\eta} \right)^{\frac{1}{2}} \delta_\beta^{2(2\beta - d + 1)} \tag{4.13}$$

simplifying into:

$$q \leq \frac{2d}{3\beta x \sqrt{y}} \sqrt{x + 2y} \delta_\beta^{2(2\beta - d + 1)}.$$

4.4.3 Further optimizations

Beyond the projection trick and the rescaling, we can apply a final standard optimization to this lattice reduction part as there is an unbalance between the size of the secret vector we want to recover and the normalized volume of the lattice. Instead of working with the full lattice coming from the descent of $\mathscr{L}$ over $\mathbb{Z}$, we can instead consider the lattice spanned by a subset of the vectors of the public basis and perform the decoding within this sublattice. The only interesting subset seems to consists in forgetting the $k \leq \frac{d}{2}$ first vectors (dropping the so-called q-vectors would not be beneficial as it would actually sparsify the lattice, making the attack worst). Doing so, the rank is of course reduced by k, at the cost of working with a lattice with covolume proportionally $q^{\frac{k}{2(d-k)}}$ bigger. The condition of eq. (4.13) updates into[12]:

$$\frac{\beta(d-k)}{(d-k)d} 3q^2 \left(x^2 y\right)^{\frac{1}{3}} \leq 2q^{\frac{n}{2n-2k}} \left(\eta^2 + \frac{2}{\eta} \right)^{\frac{1}{2}} \delta_\beta^{2(2\beta - d + k + 1)}, \tag{4.14}$$

for all $k \in \left\{0, \ldots, \frac{d}{2}\right\}$, which in turns simplifies in:

$$q \leq \min_{0 \leq k \leq \frac{n}{2}} \left(\frac{2d}{3\beta x \sqrt{y}} \sqrt{x + 2y} \delta_\beta^{2(2\beta - d + 1)} \right)^{\frac{2n-2k}{n-2k}}.$$

[11] The factor 2 accounting here for the normalized discriminant of the totally real subfield

[12] This assumes the coefficients of s are balanced, which is a reasonable assumption after the rescaling by η.

Table 4.1: Practical parameter selection

	Antrag–512	Antrag–1024
Modulus q	12289	12289
Quality α	1.17	1.63
Relative margin ε	0.08	0.30
Expected repetitions	4.2	1.0
Bit security (C/Q)	123/118	256/232
Verification key size (bytes)	896	1920
Signature size (bytes)	666	1290

The right-hand-side term grows to infinity as y goes to 0, making the attack easier and easier, recovering the intuition presented *supra* that trivial annulus (i.e., knowing exactly the value of $ff^* + gg^+$) leads to a complete key recovery in polynomial time.

Remark (On other subfield type attacks and related).

- We can also approach the problem as solving a *noisy-ring SIS* instance (namely $(1 + H)F = N + E \pmod q$) or as solving a NTRU instance with a hint, in the spirit of [DDGR20]). In both cases, we are *in fine* decoding a lattice point at distance $\|E\|$ inside a lattice of normalized volume comparable to q. Up to some minor unessential constants, all three approaches give comparable results.

- It could be tempting to go further and try projection to other subfields, but the ratio secret size to normalized volume is increasing, making the attack worse and worse, indicating that we shall only focus on the plain NTRU and on the totally real subfield.

4.4.4 Practical security assessment

This analysis translates into concrete bit-security estimates following the methodology of NewHope [ADPS16], sometimes called "core-SVP methodology". In this model [BDGL16], the bit complexity of lattice sieving (which is asymptotically the best SVP oracle) is taken as $\lfloor 0.292\beta \rfloor$ in the classical setting and $\lfloor 0.259\beta \rfloor$ in the quantum setting in dimension β. It appears that for $q > 80$ (we recall that for the chosen dimensions, $q = 12289$), the subfield attack is irrelevant in practice and the key recovery security is only driven by the first attack, directly on the original NTRU lattice. Using this analysis, we can tailor the radius α of the final annulus to match the desired security level (NIST-I and NIST-V).

4.5 Implementation and comparison

We have implemented our trapdoor generation algorithm Antrag as well as the resulting complete signature scheme in portable C based on the source codes of Falcon and Mitaka. The code archive is attached as supplementary material to this submission.

Since the signature scheme arising from Antrag is essentially identical to Mitaka for signing and verification, we largely reuse the code of Mitaka for those parts. Key generation consists of the original algorithm presented in this work to generate the first basis vector (f, g), along with code to solve the NTRU equation in order to deduce (F, G), for which we basically reuse the code of Falcon, which follows the techniques presented in [PP19]. The Fast Fourier transform and the resulting code for arithmetic in the ring are similarly borrowed from Falcon.

We note that, since the C code of Mitaka itself did not include a key generation algorithm (only precomputed fixed keys obtained using separate Python scripts), our implementation constitutes, to the best of our knowledge, the first full C implementation of a hybrid sampler-based signature.

In view of the simplicity of our trapdoor generation, the code is fairly straightforward. In particular, since the floating point uniform distributions we generate for the absolute values of the embeddings are bounded away from zero, there is no subtlety related to precision loss for values close to zero (this is unlike the Box–Muller algorithm using in signing, for which we reuse Mitaka's code that behaves properly in that respect). The only trick worth mentioning is a check in the generation of (f, g) which rejects early the pairs such that the cyclotomic integer prime above 2 divides both f and g (this is a necessary condition for the later computation of F and G to succeed, so it saves some time to test it early).

Dimension 512 and 1024 are supported, with the parameters of Table 4.1. For our trapdoor generation algorithm (as well as for signing and verification), it would not be difficult to add support other base rings (such as the 3-powersmooth cyclotomics considered in Mitaka) to reach intermediate dimensions. However, suitably optimized FFT code would be needed for those intermediate rings, and more importantly, the NTRUSolve code of [PP19] would need to be adapted as well. Neither of those steps are difficult in principle, but they represent a serious engineering effort left as future work.

A performance comparison with Falcon and Mitaka is provided in Table 4.2. Compilation is carried out with `gcc` 11.3.0 with `-O3 -march=native` optimizations enabled. Timings are collected on a single core of an Intel Core i7-7820X @ 3.60 GHz desktop machine with hyperthreading and frequency scaling disabled. Cycle counts are not provided for Falcon, since the Falcon benchmarking tool only measures clock time. The Mitaka implementation does not include a key generation procedure, explaining the missing data as well.

The running time of our key generation is very close to Falcon. Signing speeds

Table 4.2: Performance comparison with Falcon and Mitaka.

	Falcon [PFH+22]		Mitaka [EFG+22]		This work	
Dimension	512	1024	512	1024	512	1024
Quality α	1.17	1.17	2.04	2.33	1.17	1.63
Classical sec.	123	284	102	233	123	256
Key size (bytes)	896	1792	896	1792	896	1920
Sig. size (bytes)	666	1280	713	1405	666	1290
`keygen` speed (Mcycles)	—	—	N/A	N/A	27.1	83.4
`keygen` speed (ms)	6.8	19.5	N/A	N/A	7.5	23.2
`sign` speed (kcycles)	—	—	567	1109	544	1073
`sign` speed (μs)	317	635	158	309	152	299
`verif` speed (kcycles)	—	—	121	243	101	246
`verif` speed (μs)	27	61	34	68	29	69

are basically identical to Mitaka since we mostly reuse that code (up to very minor optimizations). Verification is consistent across all three schemes.

Chapter 5

Conclusion

The two aspects of cryptography, namely *constructions* and *cryptanalysis*, complement each other in the sense that cryptanalysis helps understanding the security of constructions and better constructions help develop new cryptanalysis techniques.

Currently deployed public-key cryptosystems (e.g., RSA, DSA, ECDSA) are based on the conjectured hardness of integer factorization problem or the discrete logarithm problem. However, all of these problems can be easily solved on a quantum computer running Shor's algorithm. Therefore, it is important to design new cryptographic schemes that are still secure on a quantum computer, which we usually call *post-quantum cryptography*. Among all the candidates, *lattice-based cryptography* is the most promising one because of its efficiency, strong security guarantee and versatile applications.

In Chapter 2, in an aspect that is both constructive and destructive, we analyze the hardness of binary error LWE. The standard LWE use Gussian distribution as the error distribution, but in practice, it is not very easy to implement Gussian distribution efficiently. Therefore, it is quite meaningful to analyze the hardness of LWE with respect to some other error distributions (e.g., binary error distribution). On the one hand, we analyze the complexity of binary error LWE and get a sample-time trade-off for binary error LWE. We propose a method that we call Macaulay matrix method to attack binary error LWE with less than $\Theta(n^2)$ samples. In particular, we show that, for any $\epsilon > 0$, binary error LWE can be solved in polynomial time $n^{O(1/\epsilon)}$ given $\epsilon \cdot n^2$ samples. Similarly, it can be solved in subexponential time $2^{\tilde{O}(n^{1-\alpha})}$ given $n^{1+\alpha}$ samples, for $0 < \alpha < 1$. On the other hand, we propose a variant of binary error LWE, which we call non-uniform binary error LWE. When the number of samples is strongly restricted, we prove that non-uniform binary error LWE is as hard as worst-case lattice problems. When the number of samples is not so strongly restricted and the error rate is relatively low, we propose a simple algorithm to attack non-uniform binary error LWE. These hardness results can be useful when considering the parameter setting for some cryptographic schemes based on binary error LWE.

In Chapter 3, in a purely destructive aspect, we study lattice attacks on EC(DSA). Prior to our work, lattice attacks are generally all-or-nothing, which means that if we succeed, we get the full signing key, but if we fail, we get nothing. By comparison, Bleichebacher attacks recover some bits of the signing key at each iteration. Inspired by this, we propose new ways of improving lattice attacks: *guess* some bits of the signing key, and by modifying the lattice structure, solve the resulting easier lattice problems. Interestingly, this approach is easy to simulate and parallelize, which makes the estimate of computation cost easy. Besides, the fact that numerous lattice reductions are carried out on the same lattice allows us to apply batch-CVP or CVP with preprocessing techniques, which can further improve the attack. As additional contributions, we propose variants of the attack: guessing bits of nonces and filtering signatures. Finally, we apply our ideas to attacking the TPM-Fail dataset and get an improved exploitation.

In Chapter 4, in a purely constructive sense, we study how to construct secure and efficient lattice-based signatures. In 1997, the GGH signature was proposed as the first candidate of lattice-based signature. The main idea of GGH is to use some "good" basis of some lattice as the secret key and use some "bad" basis of the same lattice as the public key. However, GGH was finally completely broken by statistical techniques, mainly because the signatures reveal information about the secret key. In 2008, the GPV framework was proposed to make hash-and-sign signatures provably secure. In 2014, Ducas, Lyubashevsky and Prest (DLP) instantiated the GPV signatures over NTRU lattices, thus making it more compact. The NIST standardization post-quantum signature scheme Falcon, is essentially a combination of DLP and FFO sampler proposed by Ducas and Prest. Very Recently, Espitau et al. proposed the Mitaka signature in order to tackle several drawbacks of Falcon. However, the Mitaka signature is less efficient and secure than Falcon. We propose a new technique to generate trapdoor basis for Mitaka: the resulting scheme is as secure as Falconand very easy to parallelize, protect against side-channel attacks. To summarize, our new scheme combines all the advantages of Falconand Mitaka with none of the drawbacks.

Appendix A

Experimental data

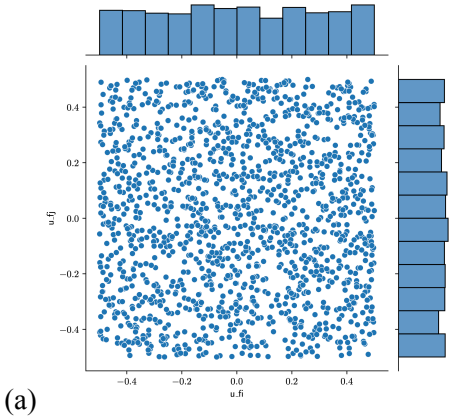
As stated in Section 4.3, we assume that each coefficient of e_f and e_g behaves like indepedent uniform random variables in $[-1/2, 1/2)$, and the squared norm of one embedding of (e_f, e_g) follows chi-squared distribution with degree of freedom 4 up to a scaling factor.

This supplementary material collects data aimed at justifying these heuristics, in the form of the following figures.

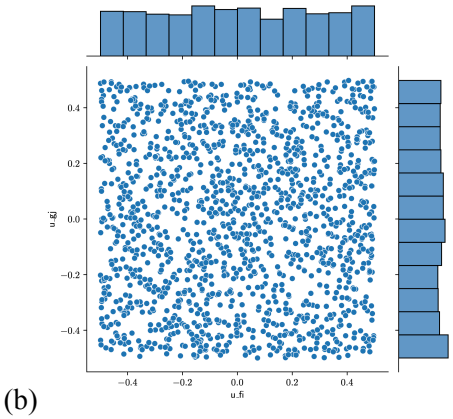
Figure A.1 shows that each coefficient of e_f and e_g do behave as independent and uniform in $[-1/2, 1/2)$. The first figure (a) shows that two randomly chosen coefficients of e_f are indeed indepedent and uniform in $[-1/2, 1/2)$. The second figure (b) shows that one randomly chosen coefficient of e_f and one randomly chosen coefficient of e_g are also indepedent and uniform in $[-1/2, 1/2)$. As a result, all the coefficients of e_f and e_g are indepedent and uniform in $[-1/2, 1/2)$.

Figure A.2 shows that the error magnitude on one embedding has the expected distribution (namely, a scaled $\chi(4)$) in the power-of-two cyclotomic case. The predicted density curve represents the true distribution of a scaled $\chi(4)$ and the experimental density curve represents the the error magnitude on one embedding. These two curves match well. Figure A.3 illustrates the similar situation of 3-powersmooth base fields.

For these experiments, 1500 samples are used, which actually can be verified by standard statistical distribution test methods (e.g., chi-squared test).



(a)



(b)

Figure A.1: Empirical joint distributions of two randomly coefficients of e_f (resp. a randomly chosen coefficient of e_f and another of e_g). The data is collected from 1500 samples (f, g) of degree $d = 512$.

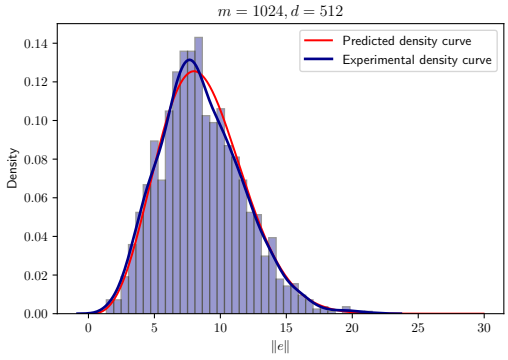


Figure A.2: Statistical density of $\|\varphi_i(e)\|$ in case $m = 1024, d = 512$ (1500 samples).

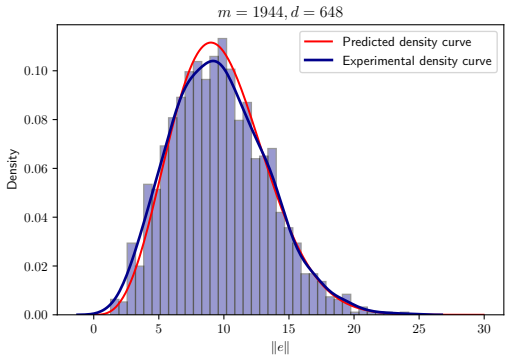


Figure A.3: Statistical density of $\|\varphi_i(e)\|$ in case $m = 1944, d = 648$ (1500 samples).

Appendix B

Publication List

B.1 Publications

- Chao Sun, Thi Thu Quyen Nguyen, Thomas Espitau, Alexandre Wallet, Mehdi Tibouchi, Masayuki Abe, "Antrag: Annular NTRU Trapdoor Generation", submitted to Public Key Cryptography 2023 (PKC 2023) (under review).
- Chao Sun, Thomas Espitau, Mehdi Tibouchi, Masayuki Abe, "Guessing Bits: Improved Lattice Attacks on (EC)DSA with Nonce Leakage", IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES), Volume 2022, Issue 1.
- Chao Sun, Mehdi Tibouchi, Masayuki Abe, "Revisiting the Hardness of Binary Error LWE", The 27th Australasian Conference on Information Security and Privacy (ACISP 2020).

B.2 Talks

- "Guessing Bits: Improved Lattice Attacks on (EC)DSA with Nonce Leakage", CHES 2022, (Leuven, Belgium).
- "Optimal Lattice Trapdoor for the Klein-GPV and Peikert Sampler", 2022 Symposium on Cryptography and Information Security. (Osaka, Japan)
- "Towards Improving Lattice Attacks on (EC)DSA", 2021 Symposium on Cryptography and Information Security. (Online)
- "Revisiting the Hardness of Binary Error LWE", ACISP 2020 (Online)

- "On the hardness of LWE with Non-Uniform Binary-Error", 2020 Symposium on Cryptography and Information Security. (Kochi, Japan)
- "Sample-Time Trade-off for the Arora-Ge Attack on Binary-Error LWE", 2019 Symposium on Cryptography and Information Security. (Otsu, Japan)

Chapter 2 is based on "Revisiting the Hardness of Binary Error LWE". In: Liu, J., Cui, H. (eds) Information Security and Privacy. ACISP 2020. Lecture Notes in Computer Science(), vol 12248. Springer, Cham. https://doi.org/10.1007/978-3-030-55304-3_22

Chapter 3 is based on "Guessing Bits: Improved Lattice Attacks on (EC)DSA with Nonce Leakage". IACR Transactions on Cryptographic Hardware and Embedded Systems, 2022(1), 391–413. https://doi.org/10.46586/tches.v2022.i1.391-413

Bibliography

[ACF+14] Martin Albrecht, Carlos Cid, Jean-Charles Faugere, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for lwe problems. 2014.

[ADH+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 717–746. Springer, Heidelberg, May 2019.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

[AFG14a] Martin R. Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving LWE by reduction to unique-SVP. In Hyang-Sook Lee and Dong-Guk Han, editors, *ICISC 13*, volume 8565 of *LNCS*, pages 293–310. Springer, Heidelberg, November 2014.

[AFG+14b] Diego F. Aranha, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, Mehdi Tibouchi, and Jean-Christophe Zapalowicz. GLV/GLS decomposition, power analysis, and attacks on ECDSA signatures with single-bit nonce bias. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 262–281. Springer, Heidelberg, December 2014.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *International Colloquium on Automata, Languages, and Programming*, pages 403–415. Springer, 2011.

[AH21a] Martin R. Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. In Anne Canteaut and François-Xavier Standaert, editors,

EUROCRYPT 2021, Part I, volume 12696 of *LNCS*, pages 528–558. Springer, Heidelberg, October 2021.

[AH21b] Martin R Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 528–558. Springer, 2021.

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

[Ajt06] Miklós Ajtai. Generating random lattices according to the invariant distribution. *Draft of March*, 2006, 2006.

[ANT+20] Diego F. Aranha, Felipe Rodrigues Novaes, Akira Takahashi, Mehdi Tibouchi, and Yuval Yarom. LadderLeak: Breaking ECDSA with less than one bit of nonce leakage. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 225–242. ACM Press, November 2020.

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[AWHT16] Yoshinori Aono, Yuntao Wang, Takuya Hayashi, and Tsuyoshi Takagi. Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 789–819. Springer, Heidelberg, May 2016.

[Bab86] László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[BCLA82] Bruno Buchberger, George E Collins, Rüdiger Loos, and Rudolph Albrecht. Computer algebra symbolic and algebraic computation. *ACM SIGSAM Bulletin*, 16(4):5–5, 1982.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th SODA*, pages 10–24. ACM-SIAM, January 2016.

[BFSY05] Magali Bardet, Jean-Charles Faugere, Bruno Salvy, and Bo-Yin Yang. Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In *The Effective Methods in Algebraic Geometry Conference (MEGA' 05)(P. Gianni, ed.)*, pages 1–14. Citeseer, 2005.

[BGG+16] Johannes Buchmann, Florian Göpfert, Tim Güneysu, Tobias Oder, and Thomas Pöppelmann. High-performance and lightweight lattice-based public-key encryption. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 2–9. ACM, 2016.

[BH19] Joachim Breitner and Nadia Heninger. Biased nonce sense: Lattice attacks against weak ECDSA signatures in cryptocurrencies. In Ian Goldberg and Tyler Moore, editors, *FC 2019*, volume 11598 of *LNCS*, pages 3–20. Springer, Heidelberg, February 2019.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[Ble00] Daniel Bleichenbacher. On the generation of one-time keys in dl signature schemes. In *Presentation at IEEE P1363 working group meeting*, page 81, 2000.

[BO88] Ernest F Brickell and Andrew M Odlyzko. Cryptanalysis: A survey of recent results. *Proceedings of the IEEE*, 76(5):578–593, 1988.

[BV96] Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 129–142. Springer, Heidelberg, August 1996.

[CGM19] Yilei Chen, Nicholas Genise, and Pratyay Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 3–32. Springer, Heidelberg, December 2019.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2011.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, September 1997.

[CPS+20] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. ModFalcon: Compact signatures based on module-NTRU lattices. In Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese, editors, *ASIACCS 20*, pages 853–866. ACM Press, October 2020.

[DDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 329–358. Springer, Heidelberg, August 2020.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DHMP13] Elke De Mulder, Michael Hutter, Mark E. Marson, and Peter Pearson. Using Bleichenbacher's solution to the hidden number problem to attack nonce leaks in 384-bit ECDSA. In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *LNCS*, pages 435–452. Springer, Heidelberg, August 2013.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Heidelberg, December 2014.

[DMQ13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 18–34. Springer, 2013.

[DN12] Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 433–450. Springer, Heidelberg, December 2012.

[DP15a] Léo Ducas and Thomas Prest. Fast Fourier orthogonalization. Cryptology ePrint Archive, Report 2015/1014, 2015. `https://eprint.iacr.org/2015/1014`.

[DP15b] Léo Ducas and Thomas Prest. A hybrid Gaussian sampler for lattices over rings. Cryptology ePrint Archive, Report 2015/660, 2015. https://eprint.iacr.org/2015/660.

[dt20] The FPLLL development team. fplll, a lattice reduction library, Version: 5.4.0. Available at https://github.com/fplll/fplll, 2020.

[EFG+22] Thomas Espitau, Pierre-Alain Fouque, François Gérard, Mélissa Rossi, Akira Takahashi, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Mitaka: A simpler, parallelizable, maskable variant of falcon. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 222–253. Springer, Heidelberg, May / June 2022.

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1857–1874. ACM Press, October / November 2017.

[EK20] Thomas Espitau and Paul Kirchner. The nearest-colattice algorithm. Cryptology ePrint Archive, Report 2020/694, 2020. https://eprint.iacr.org/2020/694.

[ETWY22] Thomas Espitau, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Shorter hash-and-sign lattice-based signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 245–275. Springer, Heidelberg, August 2022.

[Fau99] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases (f4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[FKT+20] Pierre-Alain Fouque, Paul Kirchner, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Key recovery from Gram-Schmidt norm leakage in hash-and-sign signatures over NTRU lattices. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 34–63. Springer, Heidelberg, May 2020.

[Gal12] Steven D. Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, 2012.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131. Springer, Heidelberg, August 1997.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 31–51. Springer, Heidelberg, April 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GS02] Craig Gentry and Michael Szydlo. Cryptanalysis of the revised NTRU signature scheme. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 299–320. Springer, Heidelberg, April / May 2002.

[Hen20] Nadia Heninger. Using lattices for cryptanalysis, 2020. `https://simons.berkeley.edu/talks/using-lattices-cryptanalysis`.

[HGS01] Nick A Howgrave-Graham and Nigel P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, 2001.

[HHP+03] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Heidelberg, April 2003.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.

[JSSS20] Jan Jancar, Vladimir Sedlacek, Petr Svenda, and Marek Sys. Minerva: The curse of ECDSA nonces. *IACR TCHES*, 2020(4):281–308, 2020. `https://tches.iacr.org/index.php/TCHES/article/view/8684`.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987.

[KEF21] Paul Kirchner, Thomas Espitau, and Pierre-Alain Fouque. Towards faster polynomial-time lattice reduction. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 760–790, Virtual Event, August 2021. Springer, Heidelberg.

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th SODA*, pages 937–941. ACM-SIAM, January 2000.

[LDK+22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014.

[LLL+82] Hendrik Willem Lenstra, Arjen K Lenstra, L Lovfiasz, et al. Factoring polynomials with rational coeficients. 1982.

[LN13] Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In Ed Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 293–309. Springer, Heidelberg, February / March 2013.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

[Mic10] Daniele Micciancio. Duality in lattice cryptography. In *Public key cryptography*, page 2, 2010.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of lwe search-to-decision reductions. In *Annual Cryptology Conference*, pages 465–484. Springer, 2011.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *Advances in Cryptology–CRYPTO 2013*, pages 21–39. Springer, 2013.

[MSEH20] Daniel Moghimi, Berk Sunar, Thomas Eisenbarth, and Nadia Heninger. TPM-FAIL: TPM meets timing and lattice attacks. In Srdjan Capkun and Franziska Roesner, editors, *USENIX Security 2020*, pages 2057–2073. USENIX Association, August 2020.

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 820–849. Springer, Heidelberg, May 2016.

[NR06] Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288. Springer, Heidelberg, May / June 2006.

[NS02] Phong Q. Nguyen and Igor Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15(3):151–176, June 2002.

[NT12] Phong Q Nguyen and Mehdi Tibouchi. Lattice-based fault attacks on signatures. In *Fault Analysis in Cryptography*, pages 201–220. Springer, 2012.

[Pei10] Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 80–97. Springer, Heidelberg, August 2010.

[PFH+17] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2017. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.

[PFH+22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor

Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[PP19] Thomas Pornin and Thomas Prest. More efficient algorithms for the NTRU key generation using the field norm. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 504–533. Springer, Heidelberg, April 2019.

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

[PW11] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7, 2010.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SE94] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.

[Sha82] Adi Shamir. A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 145–152. IEEE, 1982.

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

[Sun21] Chao Sun. Source code for the algorithms in this work. https://github.com/security-kouza/Lattice-Attacks-on-EC-DSA, 2021.

[The20] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.2)*, 2020. https://www.sagemath.org.

[Tib17] Mehdi Tibouchi. Attacks on Schnorr signatures with biased nonces, 2017. https://ecc2017.cs.ru.nl/slides/ecc2017-tibouchi.pdf.

[TTA18] Akira Takahashi, Mehdi Tibouchi, and Masayuki Abe. New Bleichenbacher records: Fault attacks on qDSA signatures. *IACR TCHES*, 2018(3):331–371, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7278.

[WAT17] Yuntao Wang, Yoshinori Aono, and Tsuyoshi Takagi. An experimental study of Kannan's embedding technique for the search LWE problem. In Sihan Qing, Chris Mitchell, Liqun Chen, and Dongmei Liu, editors, *ICICS 17*, volume 10631 of *LNCS*, pages 541–553. Springer, Heidelberg, December 2017.

[Wik22] Wikipedia contributors. Hoeffding's inequality — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Hoeffding%27s_inequality&oldid=1108132131, 2022. [Online; accessed 16-January-2023].

[WWW+16] Carl A. Wesolowski, Surajith N. Wanasunara, Michal J. Wesolowski, Belkis Erbas, and Paul S. Babyn. A gamma-distribution convolution model of 99mtc-mibi thyroid time-activity curves. *EJNMMI Physics*, 31, 2016.