On the Security of Lattice-based Fiat-Shamir Signatures in the Presence of Randomness Leakage *

Yuejun Liu[1,2], Yongbin Zhou[1,2**], Shuo Sun[1,2], Tianyu Wang[1,2], Rui Zhang[1,2], and Jingdian Ming[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Science
[2] School of Cyber Security, University of Chinese Academy of Science
{liuyuejun, zhouyongbin, sunshuo, wangtianyu, r-zhang, mingjingdian}@iie.ac.cn

Abstract. Leakages during the signing process, including partial key exposure and partial (or complete) randomness exposure, may be devastating for the security of digital signatures. In this work, we investigate the security of lattice-based Fiat-Shamir signatures in the presence of randomness leakage. To this end, we present a generic key recovery attack that relies on minimum leakage of randomness, and then theoretically connect it to a variant of Integer-LWE (ILWE) problem. The ILWE problem, introduced by Bootle et al. at Asiacrypt 2018, is to recover the secret vector $\mathbf{s}$ given polynomially many samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}^{n+1}$, and it is solvable if the error $e \in \mathbb{Z}$ is not superpolynomially larger than the inner product $\langle \mathbf{a}, \mathbf{s} \rangle$. However, in our variant (we call the variant FS-ILWE problem in this paper), $\mathbf{a} \in \mathbb{Z}^n$ is a sparse vector whose coefficients are NOT independent any more, and e is related to $\mathbf{a}$ and $\mathbf{s}$ as well. We prove that the FS-ILWE problem can be solved in polynomial time, and present an efficient algorithm to solve it. Our generic key recovery method directly implies that many lattice-based Fiat-Shamir signatures will be totally broken with one (deterministic or probabilistic) bit of randomness leakage per signature. Our attack has been validated by experiments on two NIST PQC signatures Dilithium and qTESLA. For example, as to Dilithium-III of 125-bit quantum security, the secret key will be recovered within 10 seconds over an ordinary PC desktop, with about one million signatures. Similarly, key recovery attacks on Dilithium under other parameters and qTESLA will be completed within 20 seconds and 31 minutes respectively.

In addition, we also present a non-profiled attack to show how to obtain the required randomness bit in practice through power analysis attacks on a proof-of-concept implementation of polynomial addition. The experimental results confirm the practical feasibility of our method.

* This work was supported in part by National Natural Science Foundation of China (No.61632020, U1936209) and Beijing Natural Science Foundation (No. 4192067).

** Corresponding author.

Keywords: Randomness leakage attacks · Fiat-Shamir signature · Dilithium · qTESLA · ILWE · the least squares method

1 Introduction

Most cryptographic algorithms are designed under the assumption that all the sensitive values are kept hidden. However, when a cryptographic algorithm is practically used, these values may be leaked to adversaries due to implementation, communication or other reasons. Taking digital signatures as an example, leakages[3] during the signing process, including partial key exposure and partial (or complete) randomness exposure, are proved to be devastating for their security. For example, Heninger and Shacham [23] showed that the RSA secret key with small public parameters can be efficiently recovered given its 27% random bits. DSA whose key is 160-bit can be totally broken if only 2 least significant bits (LSBs) of randomness are known [27]. In this work we focus on the security of signatures in the presence of partial randomness leakage(s).

Howgrave-Graham and Smart [24] proposed the first partial randomness (i.e. nonce) leakage attack on DSA by reducing it to the closest vector problem (CVP), which can be solved using Babai's nearest plane algorithm [4] together with the LLL lattice reduction algorithm [26]. However, their attack relied on several heuristic assumptions. Later, Nguyen and Shparlinski [35] presented the first provable attack on DSA with partially known randomness bits. The main idea of their attack is mapping the leakage attack on DSA to an Hidden Number Problem (HNP) introduced in [11], which can be reduced to CVP and then solved with lattice reduction algorithms. Nguyen and Shparlinski showed that their attack can apply to DSA-like signatures, including ECDSA [36] and Schnorr's signature [41].

Considering the similarity between DSA and Fiat-Shamir signatures, it is natural for us to think about the following important issue: whether or not the randomness leakage attack in [35] applies to the Fiat-Shamir signatures [19] besides Schnorr's signature whose signatures are in the form of $z = y+sc \mod q$. Recall that in HNP we aim to recover the hidden number $\alpha \in \mathbb{F}_q$ given many known random $t \in \mathbb{F}_q$ and the l MSBs of $\alpha t \bmod q$ which denote any rational u such that $|\alpha t \bmod q - u|_q \leq q/2^{l+1}$. Suppose that the l LSBs of randomness y are leaked and $y = a + 2^l b$. Obviously, the key recovery attack of Fiat-Shamir signature given leakage a is then converted to an HNP where $t = 2^{-l}c \bmod q$ and $u = (2^{-l}(a - z) - q/2^{l+1}) \bmod q$. Hence, Fiat-Shamir signatures are vulnerable to such partial randomness leakage attacks.

In 2016, NIST announced a competition to develop standards for quantum-safe public key primitives. In the post-quantum setting, lattice-based cryptography is acknowledged as one of the most promising candidates and has gained a lot of attention. There were three lattice-based signatures in the second round of NIST post-quantum cryptography (PQC) competition, two of which follow

[3] "leakage" in our work stands for general information leaked from a cryptographic device and/or implementation, and is usually of no specific forms.

the Fiat-Shamir paradigm: Dilithium [30] and qTESLA [9]. According to the latest report [34] released on July 22, 2020, Dilithium was one of the 7 finalists (3 of which are signatures), while qTESLA did not advance to the third round due to performance issues. In contrast to the theoretical security, the security of lattice-based Fiat-Shamir signatures in the presence of randomness leakage still remains an open problem.

Now there is a natural question that whether or not partial randomness leakage attacks on Fiat-Shamir signatures based on other mathematical structures, such as the attack in [35], can apply to lattice-based Fiat-Shamir signatures. Unfortunately, the answer to this question is NO. The major reason is that the secret key of lattice-based Fiat-Shamir signatures consists of one or more polynomials with small coefficients and there is a big difference between polynomial multiplication and number multiplication, making it hard to define an HNP over lattices.

1.1 Our Contributions

In this work, we aim to investigate the security of lattice-based Fiat-Shamir signatures when randomness is partially leaked. Basically, our work consists of two logically connected parts.

The first and also the fundamental part is a theoretically sound key recovery attack against lattice-based Fiat-Shamir signatures, together with computation complexity and data complexity of the proposed attack. Specifically,

- We present a new polynomial time key recovery attack on lattice-based Fiat-Shamir signatures exploiting minimum leakage of randomness (actually only one bit per signature is needed). Our method relies on one and only one bit randomness leaking during the signing process, and applies to most known lattice-based Fiat-Shamir signatures. On the other hand, our method makes no assumptions regarding the concrete implementations of signatures (for example, whether the target implementation is in software or in hardware), nor the adversary's specific strategy to get this one randomness bit (for example, which kinds of concrete leakages can be obtained and how leakages occur in practice). Therefore, our attack method has wide applicability and actually belongs to a generic attack.
- Our attack is reduced to the Fiat-Shamir integer learning with error (FS-ILWE) problem, which is a variant of the mathematical problem ILWE [12]. Besides, we analyze the special structural properties of FS-ILWE and prove it can be solved using least squares method in polynomial time. By establishing a deep connection between our attack and a mathematical problem, a solid foundation for ensuring the theoretical correctness of our attack is laid.
  We investigate the quantitative relationship among the leakage position, the leakage probability[4] and the data complexity of our key recovery attack.

[4] The leakage probability corresponds to the success rate of the randomness bit recovery in SCAs.

Specifically, almost four times as many signatures theoretically are necessary if the leakage position is shifted to the left by one bit. On the other hand, as long as the leakage probability of the randomness bit required in our attack is larger than 0.5, the secret key of lattice-based Fiat-Shamir signatures can be recovered using our method.
- We choose Dilithium and qTESLA as two cases of study to verify our key recovery attack by simulated experiments. The results validate the correctness of our method and the corresponding theoretical analysis, and also show the extremely low requirements for our attack in terms of the processing power and computing resources of computing equipment. For example, when one specific bit of randomness per signature is leaked, it takes only several seconds to recover the secret key of Dilithium on an ordinary desktop, while it takes about thousands of seconds for qTESLA[5].

The second part is supportive, aiming to provide a demonstrative case of how to obtain this required one bit of randomness in practice. Specifically, we perform a practical power analysis attack against a proof-of-concept implementation of the basic operation in lattice-based Fiat-Shamir signatures (i.e. polynomial addition) on MCU 8051 STC89C58RD+. The attack results show that when our method is put into practice, the required randomness bit can be obtained even with the help of well-known power analysis attacks. This proof-of-the-concept real-world attack initially demonstrates the practicality of the fundamental method proposed in the first part.

1.2 An Intuitive Idea of Our Attack

Our attack stems from an observation that the Fiat-Shamir signatures over lattices look like ILWE samples. Specifically, the lattice-based Fiat-Shamir signature is computed as $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, where $\mathbf{s}, \mathbf{y}, \mathbf{c}, \mathbf{z}$ are elements over the ring[6] $\mathcal{R} = \mathbb{Z}[x]/(x^N + 1)$. Considering each coefficient of $\mathbf{z}$, we have $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ where z and y are the corresponding coefficient of $\mathbf{z}$ and $\mathbf{y}$, and $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ is the corresponding coefficient of the polynomial multiplication $\mathbf{sc}$ and $\bar{\mathbf{c}}$ is a row of the rotation matrix $\mathbf{C}$ of $\mathbf{c}$. Take $\bar{\mathbf{c}}$ as the random vector $\mathbf{a}$ and y as the error e, each coefficient of the signature $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ seems like a sample of the ILWE problem. As shown in [12], the ILWE problem can be solved with high probability by the least squares method followed by rounding if the standard deviation σ_e of the error distribution χ_e is not superpolynomially larger than the standard deviation σ_a of χ_a. Generally speaking, the larger the ratio of σ_e and σ_a, the more samples we need to recover $\mathbf{s}$. However, in the lattice-based Fiat-Shamir signatures, $\bar{\mathbf{c}}$ is a sparse vector whose non-zero coefficient is either 1 or -1, and y is much larger than $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, making the ratio of σ_e and σ_a very large.

[5] The leakage position considered is $l + 1$.

[6] If a Fiat-Shamir signature scheme is based on a lattice, the secret key $\mathbf{S}$ is a matrix over $\mathbb{Z}$, and $\mathbf{y}, \mathbf{c}, \mathbf{z}$ are vectors over $\mathbb{Z}$. If a scheme is based on a module lattice (e.g. Dilithium), $\mathbf{s}, \mathbf{y}, \mathbf{z}$ are elements over $\mathcal{R}^l$ (l is a positive integer parameter) and $\mathbf{c}$ is an element over $\mathcal{R}$.

Worse still, the fatal reason why the idea does not work is that the lattice-based Fiat-Shamir signatures are filtered by the rejection sampling technique, which provides that z is independent of the secret key $\mathbf{s}$ statistically, and we cannot infer any information of $\mathbf{s}$ from z.

To overcome these technical hurdles, our approach is to establish the dependency between the signature z and the secret key $\mathbf{s}$ by leaking the randomness while keeping the signature form unchanged, and then further process the signature to reduce the difficulty of this special LWE problem. With these steps, recovering the secret the secret key $\mathbf{s}$ is exactly solving the FS-ILWE problem, in which coefficients of $\mathbf{a}$ are not mutually independent and e is related to $\mathbf{a}$ and $\mathbf{s}$ due to the rejection sampling technique. Finally, we prove that FS-ILWE can also be solved using least squares regression in polynomial time. Similar methods are also applicable to scenarios where adversaries leak randomness with a certain probability.

1.3 Related Work

Leakage Attacks on (EC)DSA. The work [24,35,36] has shown that attacks on DSA-like signature schemes with partial known randomness can be mapped to an HNP problem, which can be reduced to CVP and solved by lattice reduction techniques [4,26,16]. Based on the idea, a series of work estimated the security of implementations of DSA and ECDSA in OpenSSL [14,1,15,8,42,2]. Almost all of them used the cache-based side-channel attacks (SCAs) to extract the leaked information except [15], which used a remote timing attack to obtain the MSBs of the ECDSA randomness. [27] reduced the number of required LSB of randomness for 160-bit DSA key from 3 to 2 by proposing a new lattice reduction technique.

Leakage Attacks on Lattice-based Fiat-Shamir Signatures. Since lattice-based cryptography has received widespread attention, a large number of schemes and implementations have emerged. More recently, researchers started to investigate the implementation security against leakage and fault attacks. In the case of randomness leakage attacks on lattice-based signatures, the target of existing work on BLISS [20,18] and BLISS-B [37], both of which follow the Fiat-Shamir paradigm, is the Gaussian sampling algorithm used to generate the randomness polynomial $\mathbf{y}$. The main idea is leaking almost the entire $\mathbf{y}$ exploiting the non-constant time property of Gaussian sampling algorithm by the FLUSH+RELOAD cache-attack or the branch tracing technique. With the signature $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, the secret key $\mathbf{s}$ can be recovered via basic linear algebra or lattice reduction techniques. In contrast, our attack requires only one (deterministic or probabilistic) bit of leakage per signature, and one bit of leakage is easier to obtain in practice. Furthermore, there are no requirements for the distribution of randomness in our attack, that is, our attack is applicable to lattice-based Fiat-Shamir signatures whose randomness follows both Gaussian distribution (such as [5]) and uniform distribution (such as Dilithium and qTESLA). To be emphasized, compared to leaking randomness by SCAs, our

randomness leakage attack is more concerned on mathematical techniques to exploit randomness leakage to recover the secret key.

In the case of secret key leakage attacks on lattice-based signatures, there are two leakage sources: the rejection sampling algorithm and the polynomial multiplication. The former can be used to obtain an exact quadratic function of the secret key and a noisy linear function of the secret key using electromagnetic analysis (EMA) or branch tracing [18]. They showed how to exploit the quadratic leakage to compute the secret key, however the method can only apply to a small fraction (around 7%) of keys. [12] found that the linear leakage function can be seen as an ILWE problem, which can be solved by least squares regression, and the method applies to 100% of keys. Note that although our attack is mapped to the ILWE problem variant, our attack is totally different from that in [12]. Their attack needs to obtain a noisy linear function of the secret key by SCAs on the non-constant time rejection sampling algorithm, while our attack is a generic leakage attack and works as long as a single specific bit of randomness per signature leaks during the use process of lattice-based Fiat-Shamir signatures, without limiting the leakage methods and leakage sources. Moreover, their attack does not apply to Dilithium and qTESLA because of the uniform distribution.

Besides, side-channel vulnerabilities of polynomial multiplication have been presented, including differential power analysis (DPA) on the sparse polynomial multiplication and the schoolbook polynomial multiplication in BLISS [18] and Dilithium [40], and template attack (TA) on the Number Theoretic Transform (NTT) polynomial multiplication in lattice-based PKE [38]. However, DPA belongs to non-profiled attacks, which is less powerful than other profiled attacks and needs much more traces to successfully recover the secret key. TA is more powerful, but requires the adversary to fully control a profiling device to access a large number of profiling traces. Thus, we do not think these two attacks are practical enough for lattice-based cryptographic implementations.

2 Preliminaries

In this section, we present some basic notations and definitions.

Notations. For $x \in \mathbb{R}$, rounding the number x is denoted by $\lceil x \rfloor$. We denote column vectors and matrices in bold, respectively by bold lowercase (e.g. $\mathbf{x}$) and uppercase (e.g. $\mathbf{A}$). The Euclidean norm of the vector $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is denoted by $\|\mathbf{x}\|_2$, and the infinity norm by $\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|)$.

For any random variable X, $\mathbb{E}[X]$ denotes the expectation of X and $\mathbb{D}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ denotes the variance. We write $X \sim \chi$ to denote that X follows the distribution χ. If χ is a discrete distribution over some countable set S, then for any $s \in S$, we denote by $\chi(s)$ the probability that a sample from χ equals to s. In particular, if $f : S \to \mathbb{R}$ is any function and $X \sim \chi$, we have: $\mathbb{E}[f(s)] = \sum_{s \in S} f(s) \cdot \chi(s)$.

For the rest of the paper, we will work in the ring $\mathcal{R} \triangleq \mathbb{Z}[x]/(x^n + 1)$ where n is a power-of-two integer. For an element $\mathbf{a} = \sum_{i=0}^{n-1} a_i x^i \in \mathcal{R}$, it can also be

represented as a vector $(a_0, a_1, \ldots, a_{n-1})$. For two polynomials $\mathbf{a}, \mathbf{b}$, the inner product is denoted by $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=0}^{n-1} a_i b_i = \mathbf{a}^T \mathbf{b}$. The polynomial multiplication is represented as $\mathbf{ab}$ and can also be denoted as matrix multiplication $\mathbf{Ab}$ or $\mathbf{Ba}$ where $\mathbf{A}$, $\mathbf{B}$ are the rotation matrices related to $\mathbf{a}$ and $\mathbf{b}$. The rotation matrix $\mathbf{A}$ of $\mathbf{a}$ is the following Toeplitz matrix:

$$\mathbf{A} = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ -a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1 & -a_2 & -a_3 & \cdots & a_0 \end{bmatrix} \tag{1}$$

For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the operator norm $\|\mathbf{A}\|_p^{\text{op}}$ of $\mathbf{A}$ with respect to the p-norm is given by

$$\|\mathbf{A}\|_p^{\text{op}} = \sup_{x \in \mathbb{R}^n \setminus 0} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \sup_{\|x\|_p = 1} \|\mathbf{Ax}\|_p.$$

For $a \in \mathbb{Z}$ and $l \in \mathbb{N}$, $[a]_{2^l}$ is the l least significant bits of a in $(-2^l, 2^l)$ such that $[a]_{2^l} = a \pmod{2^l}$ when $a \geq 0$ and $[a]_{2^l} = -(|a| \pmod{2^l})$ when $a < 0$. We extend the definition to vectors: for $\mathbf{v} = (v_1, \ldots, v_n)$, $[\mathbf{v}]_{2^l}$ denotes the same length vector with entries $[v_i]_{2^l}$.

2.1 Subgaussian Distribution

In this section, we recall the notion of subgaussian distributions in [12] and collect some properties of subgaussian distributions.

Definition 1 (Subgaussian). *A random variable X over $\mathbb{R}$ is said to be τ-subgaussian for some τ if the following bound holds for all $s \in \mathbb{R}$:*

$$\mathbb{E}[\exp(sX)] \leq \exp(\frac{\tau^2 s^2}{2}).$$

Lemma 1. *A τ-subgaussian random variable X satisfies:*

$$\mathbb{E}(X) = 0 \quad \textit{and} \quad \mathbb{E}(X^2) \leq \tau^2.$$

Lemma 2. *Any distribution over $\mathbb{R}$ of mean zero and supported over a bound interval $[a, b]$ is $\frac{(b-a)}{2}$-subgaussian.*

Similar to Gaussian distributions, the tail of a subgaussian variable can be bounded.

Lemma 3. *Let X be a τ-subgaussian distribution. For any $t > 0$,*

$$\Pr[X > t] \leq \exp(-\frac{t^2}{2\tau^2}).$$

Besides, a linear combination of independent subgaussian random variables is also subgaussian.

Lemma 4. *Let $X_1, \ldots, X_n$ be independent random variables such that X_i is τ_i-subgaussian. For all $\mu_1, \ldots, \mu_n \in \mathbb{R}$, the random variable $X = \mu_1 X_1 + \cdots + \mu_n X_n$ is τ-subgaussian with:*

$$\tau^2 = \mu_1^2 \tau_1^2 + \cdots + \mu_n^2 \tau_n^2.$$

The definition of subgaussian distributions can be extended to vectors.

Definition 2. *A random vector $\mathbf{x} \in \mathbb{R}^n$ is called a τ-subgaussian random vector if for all vectors $\mathbf{u} \in \mathbb{R}^n$ with $||\mathbf{u}||_2 = 1$, the inner product $\langle \mathbf{u}, \mathbf{x} \rangle$ is a τ-subgaussian random variable.*

It is obviously that if $X_1, \ldots, X_n$ are independent τ-subgaussian random variables, then the random vector $\mathbf{x} = (X_1, \ldots, X_n)$ is τ-subgaussian, and vice versa. A nice feature of subgaussian random vectors is that the image of a random vector $\mathbf{x}$ under any linear transformation $\mathbf{A} \in \mathbb{R}^{m \times n}$ is also subgaussian. It should be emphasized that $\mathbf{Ax}$ is still subgaussian even when the distribution of $\mathbf{x}$ is related to $\mathbf{A}$, because every coefficient $\langle \mathbf{a}_i, \mathbf{x} \rangle$ of $\mathbf{Ax}$ is subgaussian according to Lemma 4, which holds as long as $x_1, \ldots, x_n$ are independent subgaussian random variables, without the necessity of independence between $\mathbf{a}_i$ and $\mathbf{x}$[7].

Lemma 5. *Let $\mathbf{x}$ be a τ-subgaussian vector in $\mathbb{R}^n$ given $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then the random vector $\mathbf{y} = \mathbf{Ax}$ is τ'-subgaussian where $\tau' = \|\mathbf{A}^T\|_2^{\text{op}} \cdot \tau$.*

Besides, extending the tail property to higher dimensions, we have the following lemma:

Lemma 6. *Let $\mathbf{v}$ be a τ-subgaussian random vector in $\mathbb{R}^n$. Then:*

$$\Pr[\|\mathbf{v}\|_\infty > t] \leq 2n \cdot \exp(-\frac{t^2}{2\tau^2}).$$

2.2 The Integer LWE Problem

A main tool of our attack is the ILWE problem, which is defined in [12] and is computed over $\mathbb{Z}$ rather than $\mathbb{Z}/q\mathbb{Z}$.

Definition 3 (ILWE Distribution). *For any vector $\mathbf{s} \in \mathbb{Z}^n$ and any two probability distribution χ_a, χ_e over $\mathbb{Z}$, the ILWE distribution $\mathcal{D}_{\mathbf{s},\chi_a,\chi_e}$ associated with those parameters is the probability distribution over $\mathbb{Z}^n \times \mathbb{Z}$ defined as follows: samples from $\mathcal{D}_{\mathbf{s},\chi_a,\chi_e}$ are of the form*

$$(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$$

where $\mathbf{a} \leftarrow \chi_a^n$ and $e \leftarrow \chi_e$.

Definition 4 (ILWE Problem). *Given m samples $\{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq m}$ from the ILWE distribution $\mathcal{D}_{\mathbf{s},\chi_a,\chi_e}$ for some $\mathbf{s} \in \mathbb{Z}^n$, one is asked to recover the vector $\mathbf{s}$.*

[7] For completeness, we provide proofs of Lemma 4 and Lemma 5 in Appendix A, which are almost the same as that in [12].

Let σ_e and σ_a be the standard deviation of the error distribution χ_e and the coefficient distribution χ_a respectively. Bootle et al. [12] showed the ILWE problem with m samples can be solved in polynomial time using statistical learning techniques when $m \geq \Omega(\sigma_e/\sigma_a)^2$ and σ_e is not superpolynomially larger than σ_a.

3 The Partial Randomness Leakage Attack

In the section, we present a polynomial time attack to recover the secret key of lattice-based Fiat-Shamir signatures with minimum leakage of randomness.

In a Fiat-Shamir signature scheme whose form of signature is $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, the random oracle output $\mathbf{c}$ and the signature $\mathbf{z}$ are known, while the secret key $\mathbf{s}$ and the randomness $\mathbf{y}$ are unknown. Each coefficient z of $\mathbf{z}$ is obtained by $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, where $\bar{\mathbf{c}}$ is a corresponding row of the rotation matrix $\mathbf{C}$ of $\mathbf{c}$. A natural way to recover $\mathbf{s}$ is leaking the whole y and solving a system of linear equations. In the following, we show how to minimize the number of leakage bits of y. As a result, we can recover the secret key $\mathbf{s}$ even if only one bit of y is leaked per signature.

There are two crucial issues to recover $\mathbf{s}$. Firstly, the distribution of the signature z is not related to the distribution of $\mathbf{s}$ due to the rejection sampling technique, so we build the relationship between z and $\mathbf{s}$ via one randomness bit of leakage and reduce the key recovery attack to an FS-ILWE problem in Section 3.1. Another obstacle comes from solving the FS-ILWE problem, whose distributions of the random vector $\mathbf{a}$ and the error e are different from those in the ILWE problem. We show that FS-ILWE problem is also solvable with linear regression in Section 3.3.

3.1 Description of Our Attack

The crux of our attack is reducing it to a problem called FS-ILWE and in this section, we describe the reduction procedure in four steps. The whole description of our attack (leaking the $(l+1)$-th bit of y) is given in Algorithm 1.

Step 1: Throw Away the Most Significant Bits Note that in the Fiat-Shamir signature scheme, y is used to mask the value of $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ and we always pick a y in a range that is much larger than the range of $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, that is, only the low-order bits of the signature z are related to the secret key s. Therefore, there is no need to leak the whole y to recover $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ and only the least significant bits of y are necessary. For example, in Dilithium-III, $\|\mathbf{sc}\|_\infty$ is less than 6 bits[8]. In such case, we only need to leak the 6 least significant bits of y to recover $\mathbf{s}$. It should be noted that here we need to leak the extra seventh bit to recover the exact value of $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$.

[8] In Dilithium, the original β is 8 bits so that $\|\mathbf{sc}\|_\infty \leq \beta$ except with a probability of 2^{-80}. However, in practice most of $\mathbf{sc}$ is much smaller than that bound and we take 6 bits as the real bound since $\|\mathbf{sc}\|_\infty \leq 2^6$ with 99% probability according to the statistical result.

Algorithm 1 Partial Randomness Leakage Attack($\Sigma = ((\bar{\mathbf{c}}^{(1)}, z^{(1)}), \ldots, (\bar{\mathbf{c}}^{(m)}, z^{(m)})), \mathcal{Y} = (y_{l+1}^{(1)}, \ldots, y_{l+1}^{(m)})$)

for $i = 1$ to m **do**
  $\mathbf{a}^{(i)} = \bar{\mathbf{c}}^{(\mathbf{i})}$
  if $z^{(i)} > 0$ **then**
    $b^{(i)} = z^{(i)} \bmod 2^l$
  else
    $b^{(i)} = -(|z^{(i)}| \bmod 2^l)$
  end if
  if $z_{l+1}^{(i)} \neq y_{l+1}^{(i)}$ **then**
    if $-2^l < b^{(i)} < -2^{l-1}$ or $0 < b^{(i)} < 2^{l-1}$ or ($b^{(i)} == 0$ and $z^{(i)} > 0$) **then**
      $b^{(i)} = b^{(i)} + 2^l$
    else
      $b^{(i)} = b^{(i)} - 2^l$
    end if
  end if
end for
$\mathbf{A} = \begin{pmatrix} \mathbf{a}^{(1)} \\ \vdots \\ \mathbf{a}^{(m)} \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} b^{(1)} \\ \vdots \\ b^{(m)} \end{pmatrix}$
$\tilde{\mathbf{s}}$ = Least-Squares-Method($\mathbf{A}, \mathbf{b}$)
$\mathbf{s} = \lceil \tilde{\mathbf{s}} \rfloor$
return s

Step 2: Throw Away the Least Significant Bits Another difference between lattice-based Fiat-Shamir signature and Fiat-Shamir signatures based on other mathematical structures is that the former is computed without modular reduction. Taking $\bar{\mathbf{c}}$ as the random vector $\mathbf{a}$ and y as the error e, each coefficient of the Fiat-Shamir signature $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ looks like a sample of the ILWE problem. As shown in [12], such a problem can be solved in polynomial time using statistical learning technique. Combining with the first observation above, we can reduce the leakage attack on the lattice-based Fiat-Shamir signatures to an ILWE-like problem with relatively small errors. Assuming $\|\mathbf{sc}\|_\infty < 2^l$, we can rewrite the signature as:

$$z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle \tag{2}$$

$$\Rightarrow z \bmod 2^l = (y \bmod 2^l + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \pmod{2^l} \tag{3}$$

$$\Rightarrow [z]_{2^l} \pm d \cdot 2^l = [y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle \tag{4}$$

where we let

$$\mathbf{a} = \bar{\mathbf{c}}, \quad e = [y]_{2^l} \quad \text{and} \quad b = [z]_{2^l} \pm d \cdot 2^l.$$

(3) follows from the fact $\|\mathbf{sc}\|_\infty < 2^l$ and (4) follows from the leakage of y. That is, without extra information of y, we cannot remove the modulus in (4) and cannot reduce the attack to the ILWE-like problem. Hence, we need to leak

the $(l+1)$-th bit of y to judge whether the sum of $[y]_{2^l}$ and $\mathbf{sc}$ exceeds l bits. Specifically, if the $(l+1)$-th bit of y and the $(l+1)$-th bit of z are the same, then $d=0$, otherwise $d=1$.

Collecting multiple samples of the form (4), the problem of recovering the secret $\mathbf{s}$ is thus an ILWE-like problem in which the random vector $\mathbf{a}$ is the output of the random oracle with special structure and the error term $\mathbf{e}$ is not independent of $\mathbf{a}$ and $\mathbf{s}$ due to the rejection sampling. This problem is called the FS-ILWE problem in the rest of the paper. Later we will estimate the distribution of the error term, denoted by $\chi_e^{(\mathbf{a},\mathbf{s})}$.

Step 3: Determine the Sign Caused by Overflow In addition, in the case of overflow ($d=1$), we need to determine whether it is caused by a carry or a borrow – i.e. determine whether $b=[z]_{2^l}+2^l$ (carry occurs) or $b=[z]_{2^l}-2^l$ (borrow occurs). Our strategy is determining b based on the value of $[z]_{2^l}$. Roughly speaking, if $[z]_{2^l}\geq 0$, then $[y]_{2^l}\geq 0$ and if $[z]_{2^l}\leq 0$, then $[y]_{2^l}\leq 0$. Suppose $|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}$. When an overflow occurs, there are three cases[9]:

- $[z]_{2^l}>0$: b is bounded by: $-2^{l-1}<b<2^l+2^{l-1}$. That is, if $[z]_{2^l}\in(0,2^{l-1})$, carry occurs; if $[z]_{2^l}\in(2^{l-1},2^l)$, borrows occurs.
- $[z]_{2^l}<0$: Similarly, b is bounded by: $-2^l-2^{l-1}<b<2^{l-1}$. That is, if $[z]_{2^l}\in(-2^l,-2^{l-1})$, carry occurs; if $[z]_{2^l}\in(-2^{l-1},0)$, borrows occurs.
- $[z]_{2^l}=0$: if $z>0$, carry occurs; if $z<0$, borrow occurs.

Because both a carry and a borrow are possible for some values of $[z]_{2^l}$, determining the value of $[z]_{2^l}\pm 2^l$ will introduce extra errors. However, our strategy is almost always correct if $|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}$. Hence, in order to guess b correctly, $\Pr[|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}]\approx 1$ is a necessary condition and we choose l satisfying $\Pr[|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}]>99\%$ in actual experiments[10]. In general, when we launch an attack, we firstly judge whether there is an overflow, and if so, we determine the value of b according to the value of $[z]_{2^l}$: $b=[z]_{2^l}+2^l$ when $[z]_{2^l}\in(-2^l,-2^{l-1})\cup(0,2^{l-1})\cup\{0\}_{z>0}$, and $b=[z]_{2^l}-2^l$ when $[z]_{2^l}\in(-2^{l-1},0)\cup(2^{l-1},2^l)\cup\{0\}_{z<0}$. Here we heuristically assume that the guess of the sign caused by overflow is always correct.

Step 4: Estimate the Distribution $\boldsymbol{\chi_e^{(\mathbf{a},\mathbf{s})}}$ of the Error Term We now turn our attention to the error term e, which is written as $e=[y]_{2^l}=[z-\langle \mathbf{s},\bar{\mathbf{c}}\rangle]_{2^l}$. Because of the rejection sampling technique, each coefficient z of signatures is independent from the secret key $\mathbf{s}$ and follows a public and fixed distribution, denoted by χ_z, including the discrete Gaussian distribution and the uniform distribution. To facilitate understanding, we assume z follows a uniform distribution over $(-2^\gamma,2^\gamma)\cap\mathbb{Z}$, then y follows a uniform distribution over

[9] When $[z]_{2^l}=\pm 2^{l-1}$ and $|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}$, no overflow occurs.

[10] It is worth noting that in step 1 we require that l satisfying $\Pr[|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^l]>99\%$, and in step 2 the constraint condition of l is the probability of $|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}$ is larger than 99%. The final constraint we use in the experiments is the intersection of two conditions, i.e. $\Pr[|\langle \mathbf{s},\bar{\mathbf{c}}\rangle|<2^{l-1}]>99\%$.

$(-2^\gamma - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 2^\gamma - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \cap \mathbb{Z}$, denoted by $\chi_e^{(\mathbf{a},\mathbf{s})}$. Therefore the probability density function of $e = [y]_{2^l}$[11] is

$$p(x) = \begin{cases} \sum\limits_{\xi<0,\ \xi \equiv x \bmod 2^l} p_y(\xi), & x \in (-2^l, 0) \cap \mathbb{Z} \\ \sum\limits_{\xi \equiv 0 \bmod 2^l} p_y(\xi), & x = 0 \\ \sum\limits_{\xi>0,\ \xi \equiv x \bmod 2^l} p_y(\xi), & x \in [0, 2^l) \cap \mathbb{Z} \end{cases} = \begin{cases} \frac{2^{\gamma-l}}{2^{\gamma+1}-1}, & x \in (-2^l, -\langle \mathbf{s}, \bar{\mathbf{c}} \rangle] \cap \mathbb{Z} \\ \frac{2^{\gamma-l}+1}{2^{\gamma+1}-1}, & x \in (-\langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 0) \cap \mathbb{Z} \\ \frac{2^{\gamma-l+1}}{2^{\gamma+1}-1}, & x = 0 \\ \frac{2^{\gamma-l}}{2^{\gamma+1}-1}, & x \in (0, 2^l - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \cap \mathbb{Z} \\ \frac{2^{\gamma-l}-1}{2^{\gamma+1}-1}, & x \in [2^l - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 2^l) \cap \mathbb{Z} \end{cases}$$

It is easy to work out that $\mathbb{E}([y]_{2^l}) = -\frac{2^l-1}{2^{\gamma+1}-1}\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ is close to 0. For simplicity, we approximately regard the distribution of $[y]_{2^l}$ as subgaussian over a bounded interval $(-2^l, 2^l)$ and the heuristic assumption can finally be removed.

Taken together, the attack in the presence of randomness leakage is reduced to the FS-ILWE problem and we show it can be solved with $O((n\tau_e/h)^2 \log(n))$ samples using the least squares regression in Section 3.3.

Up to now, we can recover the secret key of lattice-based Fiat-Shamir signatures with only one bit leakage of the randomness per signature and the leakage is necessary for our attack as shown in (4). Another reason we cannot recover the secret key without leakage is that lattice-based Fiat-Shamir signatures $\mathbf{z}$ are filtered by the rejection sampling, which provides that $\mathbf{z}$ are independent from the secret key $\mathbf{s}$. Therefore, to some extent, the rejection sampling technique fundamentally eliminates the potential threat of statistical attacks like ours in the leak-free setting. A detailed analysis of the attack without leakage can be found in Appendix B.

3.2 High-Order Bit Leakage

We have shown how to recover the secret key with the $(l+1)$-th bit of $\mathbf{y}$ and in this section, we give a similar argument with leakage at other known position. Suppose the leakage bit is from the t-th bit of $\mathbf{y}$ where $l+1 \le t \le k$ and k is the length of coefficients of $\mathbf{y}$. Applying the leakage attack in section 3.1 to this case directly, we can get the following FS-ILWE problem:

$$[z]_{2^{t-1}} \pm d \cdot 2^{t-1} = [y]_{2^{t-1}} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle \tag{5}$$

where we let

$$\mathbf{a} = \bar{\mathbf{c}}, \quad e = [y]_{2^{t-1}} \quad \text{and} \quad b = [z]_{2^{t-1}} \pm d \cdot 2^{t-1}.$$

[11] Here we only provide the probability density function when $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle > 0$, and the case when $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle < 0$ is similar.

Compared with (4), the only difference is the error distribution in FS-ILWE. The error distribution in (4) is an approximately subgaussian distribution over $(-2^l, 2^l)$ and in (5) it can also be approximated to a subgaussian distribution but with larger bounds $(-2^{t-1}, 2^{t-1})$. Thus, we need more samples to compute the secret key with the t-th leakage bit of randomness. Roughly, whenever the leakage location is shifted to the left by one bit, then the subgaussian moment of error τ_e doubles and almost four times as many as samples are necessary.

3.3 Solving the FS-ILWE Problem

In this section we would like to show how to solve FS-ILWE using the least squares method, which is similar to that for solving ILWE.

First, we provide a definition of FS-ILWE. In FS-ILWE, the random vector is one output of the random oracle (or hash function). Specifically, in Dilithium or qTESLA, the output of the hash function is an n-dimensional vector and has h non-zero coefficients that are either -1 or 1 with equal probability. Denote the output set by B_h and the definition of FS-ILWE is given blow.

Definition 5 (FS-ILWE Distribution). *For any vector $\mathbf{s} \in \mathbb{Z}^n$, the FS-ILWE distribution $\mathcal{D}_{\mathbf{s},B_h,\chi_e^{(\mathbf{a},\mathbf{s})}}$ associated with those parameters is the probability distribution over $\mathbb{Z}^n \times \mathbb{Z}$ defined as follows: samples from $\mathcal{D}_{\mathbf{s},B_h,\chi_e^{(\mathbf{a},\mathbf{s})}}$ are of the form*

$$(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$$

where $\mathbf{a} \leftarrow B_h$ and $e \leftarrow \chi_e^{(\mathbf{a},\mathbf{s})}$.

Definition 6 (FS-ILWE Problem). *Given m samples $\{(\mathbf{a}_i, b_i)\}_{1 \le i \le m}$ from the FS-ILWE distribution $\mathcal{D}_{\mathbf{s},B_h,\chi_e^{(\mathbf{a},\mathbf{s})}}$ for some $\mathbf{s} \in \mathbb{Z}^n$, one is asked to recover the vector $\mathbf{s}$.*

Note that for simplicity, the distribution of the error term in this section is subgaussian, but it is not exactly consistent with the real attack setting, in which the distribution is $\chi_e^{(\mathbf{a},\mathbf{s})}$. In Appendix C we will provide a theoretical justification of why FS-ILWE whose error term distribution is $\chi_e^{(\mathbf{a},\mathbf{s})}$ is solvable.

The FS-ILWE equation for $\mathbf{s}$ can be written in matrix form:

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{6}$$

where $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $\mathbf{e} \in \mathbb{Z}^m$ is subgaussian.

The idea to solve $\mathbf{s}$ using the least squares method is to find an approximate solution $\tilde{\mathbf{s}} \in \mathbb{R}^n$ of the noisy linear system (6) such that the squared Euclidean norm $\|\mathbf{b} - \mathbf{A}\tilde{\mathbf{s}}\|_2^2$ is minimal. If we can establish the bound

$$\|\mathbf{s} - \tilde{\mathbf{s}}\|_\infty < 1/2 \tag{7}$$

then we can simply round $\tilde{\mathbf{s}}$ coefficient by coefficient to get $\mathbf{s} = \lceil \tilde{\mathbf{s}} \rfloor = (\lceil \tilde{s}_1 \rfloor, \ldots, \lceil \tilde{s}_n \rfloor)$ and the FS-ILWE problem is solved [12]. In particular, when m is large, $\mathbf{A}^T\mathbf{A}$ will be invertible and we can compute $\tilde{\mathbf{s}} = (\mathbf{A}^T\mathbf{A})^{-1} \cdot \mathbf{A}^T\mathbf{b}$. Therefore, we have

$$\tilde{\mathbf{s}} - \mathbf{s} = (\mathbf{A}^T\mathbf{A})^{-1} \cdot \mathbf{A}^T\mathbf{e} = \mathbf{M}\mathbf{e} \tag{8}$$

where $\mathbf{M}$ is the matrix $(\mathbf{A}^T\mathbf{A})^{-1} \cdot \mathbf{A}^T$. Since $\mathbf{e}$ is a τ_e-subgaussian vector, $\tilde{\mathbf{s}} - \mathbf{s} = \mathbf{M}\mathbf{e}$ is also τ'-subgaussian follows from Lemma 5 where

$$\begin{aligned}\tau' &= \|\mathbf{A}^T\|_2^{\mathrm{op}} \cdot \tau_e = \tau_e \sqrt{\lambda_{\max}(\mathbf{M}\mathbf{M}^T)} = \tau_e \sqrt{\lambda_{\max}((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \cdot \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1})} \\ &= \tau_e \sqrt{\lambda_{\max}((\mathbf{A}^T\mathbf{A})^{-1})} = \frac{\tau_e}{\sqrt{\lambda_{\min}(\mathbf{A}^T\mathbf{A})}}.\end{aligned}$$

Now it remains to bound the smallest eigenvalue $\lambda_{\min}(\mathbf{A}^T\mathbf{A})$ so as to satisfy the condition in (7). In the original ILWE, the coefficients of each row $\mathbf{a}_i$ of $\mathbf{A}$ follow a τ-subgaussian distribution and every coefficient of any of $\mathbf{a}_i$ is independent from all the others. When χ_a is a subgaussian distribution, the bound can be derived from a lemma [25, Lemma 2] which is a tail inequality for the smallest and largest eigenvalues of subgaussian random vectors. However, it no longer holds in our leakage attack. In our FS-ILWE, every row $\mathbf{c}$ of $\mathbf{A}$ is sampled from B_h but each row is independent of each other. Obviously, the coefficients of $\mathbf{c}$ are not independent, however, $\mathbf{c}$ has the following good properties.

Lemma 7. *Let $\mathbf{c}_1, ..., \mathbf{c}_m$ be sampled from B_h independently, then they satisfy:*

1. $\mathbb{E}[\mathbf{c}_i\mathbf{c}_i^T | \mathbf{c}_1, ..., \mathbf{c}_{i-1}] = \mathbb{E}[\mathbf{c}_i\mathbf{c}_i^T] = \frac{h}{n}\mathbf{I}$;
2. $\mathbb{E}[\exp(\boldsymbol{\alpha}^T\mathbf{c}_i) | \mathbf{c}_1, ..., \mathbf{c}_{i-1}] = \mathbb{E}[\exp(\boldsymbol{\alpha}^T\mathbf{c}_i)] \leq \exp(\frac{1}{2})$ *for all* $\boldsymbol{\alpha} \in \mathbb{R}^n$ *with* $\|\boldsymbol{\alpha}\|_2 = 1$, *and* $\mathbf{c}_i$ *is a* 1*-subgaussian random vector for all* $i = 1, .., m$.

Proof.

1. If we write $\mathbf{c}_i = (c_{i1}, ..., c_{in})$, in order to calculate $\mathbb{E}[\mathbf{c}_i\mathbf{c}_i^T]$, we need to know $\mathbb{E}[c_{ij}c_{ij}]$ and $\mathbb{E}[c_{ij}c_{ik}](j \neq k)$. For the first expectation, we have:

$$\mathbb{E}[c_{ij}c_{ij}] = \Pr[c_{ij} = 1] \cdot 1^2 + \Pr[c_{ij} = -1] \cdot (-1)^2 = \frac{h}{2n} + \frac{h}{2n} = \frac{h}{n}$$

for all $i = 1, .., m$ and $j = 1, .., n$.
Although c_{ij} and $c_{ik}(j \neq k)$ are not independent, fortunately, their covariance is 0:

$$\begin{aligned}\mathbb{E}[c_{ij}c_{ik}] &= (\Pr[c_{ij} = 1, c_{ik} = 1] + \Pr[c_{ij} = -1, c_{ik} = -1]) - (\Pr[c_{ij} = 1, c_{ik} = -1] \\ &\quad + \Pr[c_{ij} = -1, c_{ik} = 1]) \\ &= (\frac{h \cdot (h-1)}{2n \cdot 2(n-1)} + \frac{h \cdot (h-1)}{2n \cdot 2(n-1)}) - (\frac{h \cdot (h-1)}{2n \cdot 2(n-1)} + \frac{h \cdot (h-1)}{2n \cdot 2(n-1)}) = 0\end{aligned}$$

[12] The reason why FS-ILWE is solvable even when $\chi_e^{(\mathbf{a},\mathbf{s})}$ is not subgaussian is that the additional error introduced by the distribution of e is much smaller than $1/2$ and it don't affect the rounding at the end.

for all $i = 1, .., m$ and $j, k = 1, ..n$ with $j \neq k$.

2. Because every vector $\mathbf{c}_i$ from B_h has h non-zero coefficients, without loss of generality, we assume that the first h coefficients of $\mathbf{c}_i$ are non-zero, then $c_{ij}(1 \leq j \leq h)$ is a *Rademacher* random variable, and c_{ij} and $c_{ik}(1 \leq j, k \leq h,\ j \neq k)$ are independent. If we write $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)$, we have:

$$\begin{aligned}\mathbb{E}[\exp(\boldsymbol{\alpha})^T \mathbf{c}_i] &\leq \mathbb{E}[\exp(\alpha_1 c_{i1} + ... + \alpha_h c_{ih})] \leq \mathbb{E}[\exp(\alpha_1 c_{i1})]...\mathbb{E}[\exp(\alpha_{\mathrm{h}} c_{ih})] \\ &\leq \exp(\frac{\alpha_1^2}{2})...\exp(\frac{\alpha_h^2}{2}) = \exp(\frac{\alpha_1^2}{2} + ... + \frac{\alpha_h^2}{2}) = \exp(\frac{1}{2})\end{aligned}$$

for all $i = 1, .., m$. The third inequality is followed that the *Rademacher* random variable is a 1-subgaussian random variable.

In order to estimate the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$, we adapt the lemma [25, Lemma 2] to our analysis by specializing their statement to $\epsilon_0 = 1/4$ and $\gamma = \sqrt{n/h}$.

Lemma 8. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be random vectors in $\mathbb{R}^n$ such that,*

$$\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^T | \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}] = \mathbf{I} \quad \textit{and}$$

$$\mathbb{E}[\exp(\boldsymbol{\alpha}^T \mathbf{x}_i) | \mathbf{x}_1, \ldots, \mathbf{x}_{i-1}] \leq \exp(\frac{\sqrt{n}}{2\sqrt{h}}) \quad \textit{for all } \boldsymbol{\alpha} \in \mathbb{R}^n \textit{ with } \|\boldsymbol{\alpha}\|_2 = 1$$

for all $i = 1, \ldots, m$, almost surely. For any $\delta \in (0, 1)$,

$$\Pr[\lambda_{\max}(\frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T) > 1 + 2\varepsilon_{\delta,m} \quad \textit{or} \quad \lambda_{\min}(\frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T) < 1 - 2\varepsilon_{\delta,m}] \leq \delta \quad (9)$$

where $\varepsilon_{\delta,m} := 2\sqrt{\frac{n}{h}} \cdot (\sqrt{\frac{8(n\log(9)+\log(2/\delta))}{m}} + \frac{n\log(9)+\log(2/\delta)}{m})$.

If we write $\mathbf{A}^T = (\mathbf{c}_1, ...\mathbf{c}_m)$, then $\mathbf{A}^T\mathbf{A}$ can be expressed by $\sum_{i=1}^{m} \mathbf{c}_i \mathbf{c}_i^T$. Combining Lemma 7 and Lemma 8, we get the bound on the smallest eigenvalue of $\mathbf{A}^T\mathbf{A}$ by replacing $\mathbf{x}_i$ with $\sqrt{n/h} \cdot \mathbf{c}_i$ $(1 \leq i \leq m)$.

Theorem 1. *Let $\mathbf{A}$ be an $m \times n$ random matrix and every row $\mathbf{c}_i (1 \leq i \leq m)$ of it is sampled from B_h independently. There exist constants C_1, C_2 such that for all $\beta \in (0, 1)$ and $\eta \geq 1$, if $m \geq n(C_1 n + C_2\eta)/(h\beta^2)$ then*

$$\Pr[\lambda_{\max}(\mathbf{A}^T\mathbf{A}) > (1+\beta) \cdot \frac{mh}{n} \quad \textit{or} \quad \lambda_{\min}(\mathbf{A}^T\mathbf{A}) < (1-\beta) \cdot \frac{mh}{n}] < 2^{-\eta}$$

Furthermore, one can choose $C_1 = 144 \log 9$ and $C_2 = 288 \log 2$.

Proof. Let $\mathbf{x}_i = \sqrt{n/h} \cdot \mathbf{c}_i (1 \le i \le m)$. According to Lemma 7, we can easily derive that $\mathbf{x}_i$ meets the condition of Lemma 8. As $\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T = (h/n) \sum_{i=1}^{m} \mathbf{c}_i \mathbf{c}_i^T = (h/n)\mathbf{A}^T\mathbf{A}$, we plug the relation into equation (9):

$$\Pr[\lambda_{\max}(\sum_{i=1}^{m} \mathbf{A}^T\mathbf{A}) > (1+2\varepsilon_{\delta,m}) \cdot \frac{mh}{n} \quad or \quad \lambda_{\min}(\sum_{i=1}^{m} \mathbf{A}^T\mathbf{A}) < (1-2\varepsilon_{\delta,m}) \cdot \frac{mh}{n}] \le \delta \tag{10}$$

Let $\rho = (n\log(9) + \log(2/\delta))/m$ and $\delta = 2^{-\eta}$, we can simplify the expression of $\varepsilon_{\delta,m}$ to $2\sqrt{(n\rho)/h}(\sqrt{8} + \sqrt{\rho})$. If $m \ge 144n(n\log(9) + \log(2^{1+\eta}))/(h\beta^2)$, there are $\sqrt{8} + \sqrt{\rho} \le 3$, then we have:

$$2\varepsilon_{\delta,m} \le 12\sqrt{n\rho/h} \le \beta. \tag{11}$$

Equation (10) (11) with $\delta = 2^{-\eta}$ can derive our result.

Combining Theorem 1 and Lemma 6, we can bound the distance between the least squares estimator $\tilde{\mathbf{s}}$ and the actual solution $\mathbf{s}$ in the infinity norm to obtain the inequality of the form (7) with very high probability. The formal theorem is given below.

Theorem 2. *Suppose that there exists a common constant τ_e such that for all $\mathbf{a}$, $\chi_e^{(\mathbf{a},\mathbf{s})}$ is a τ_e-subgaussian vector, and $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ is sampled from the FS-ILWE distribution for some $\mathbf{s} \in \mathbb{Z}^n$ where rows of $\mathbf{A}$ are sampled from B_h independently. There exist constants $C_1, C_2 > 0$ such that for all $\eta \ge 1$, if*

$$m \ge 4n(C_1 n + C_2\eta)/h \quad and \quad m \ge 32\frac{n\tau_e^2}{h}\log(2n)$$

then the least squares estimator $\tilde{\mathbf{s}} = (\mathbf{A}^T\mathbf{A})^{-1} \cdot \mathbf{Ab}$ satisfies $\|\tilde{\mathbf{s}} - \mathbf{s}\| < 1/2$, and hence $\lfloor \tilde{\mathbf{s}} \rceil = \mathbf{s}$, with probability at least $1 - \frac{1}{2n} - 2^{-\eta}$.

Proof. Applying Theorem 1 with $\beta = 1/2$ and the same constants C_1, C_2 as introduced in the statement of that theorem, we obtain that for $m \ge 4n(C_1 n + C_2\eta)/h$, we have

$$\Pr[\lambda_{\min}(\mathbf{A}^T\mathbf{A}) < \frac{mh}{2n}] < 2^{-\eta}$$

We have shown above that $\mathbf{s} - \tilde{\mathbf{s}}$ is a $\tilde{\tau}$-subgaussian random vector with $\tilde{\tau} = \tau_e/\sqrt{\lambda_{\min}(\mathbf{A}^T\mathbf{A})}$. Applying Lemma 6 with $t = 1/2$, we have:

$$\Pr[\|\mathbf{s} - \tilde{\mathbf{s}}\|_\infty > 1/2] \le \exp(\log(2n) - \frac{mh}{16n\tau_e^2})$$

Thus, if we assume that $m \ge 32\frac{n\tau_e^2}{h}\log(2n)$, it follows that:

$$\Pr[\|\mathbf{s} - \tilde{\mathbf{s}}\|_\infty > 1/2] \le \exp(\log(2n) - 2\log(2n)) = \frac{1}{2n}.$$

It is worth noting that the cost of solving FS-ILWE problem using the least squares method equals to the complexity of computing $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ and the matrix $\mathbf{A}$ consisting of $\mathbf{c}$ is a sparse matrix, so the complexity of the problem is at most $O(h^2 \cdot m + n^3)$. It can be very efficient in practice.

3.4 Probabilistic Randomness Bit of Leakages

All the theoretical analysis above is based on an implicit assumption that an adversary is capable of obtaining the required randomness bit with certainty, which is a strict condition, because many practical issues will definitely affect the success rate of the required randomness bit on a real signature implementation. In this section we show that our attack still works with probabilistic randomness bit. We consider the leakage position to be $l+1$, and other high-order bits are similar.

Denote by κ the leakage probability of the randomness bit and $\kappa \in (0.5, 1]$. Note that $\kappa = 0.5$ means that there is no extra information about the secret key $\mathbf{s}$, thus we cannot recover $\mathbf{s}$ in this case. Similar to the case of deterministic randomness bit, we can also obtain an FS-ILWE sample b_κ thus a new FS-ILWE problem. Intuitively, for any $\kappa \in (0.5, 1]$, the expectation of $b_\kappa = [z]_{2^l} \pm d_\kappa \cdot 2^l$ should be related to the leakage probability κ, so the variable $b_\kappa - \mathbb{E}(b_\kappa)$ is also subgaussian. Next, we'll prove we can use the same method to solve this new FS-ILWE problem, and the only difference is that we must multiply the least squares estimator $\tilde{\mathbf{s}}$ by a correction coefficient before rounding to recover s in the last step. Here we also consider the distribution of $[z]_{2^l}$ as a uniform distribution over $(-2^l, 2^l) \cap \mathbb{Z}$ and $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle \in (-2^{l-1}, 2^{l-1})$.

In practice, the leakage probability κ is larger than 0.5, but according to symmetry, we extend the domain of κ to $[0, 1]$. Set the probability density function of b_κ as $\phi_\kappa = \kappa\phi_1 + (1-\kappa)\phi_0$, where ϕ_1 and ϕ_0 represents the probability density function of b_1 and b_0 respectively. When $\kappa = 1$, we can obtain all the correct samples, namely, $b_1 = [z]_{2^l} \pm d_1 \cdot 2^l = [y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle = \langle \mathbf{s}, \bar{\mathbf{c}} \rangle + e_1$. Then b_1 follows a distribution over $(-2^l + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 2^l + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \cap \mathbb{Z}$.

Another extreme case is $\kappa = 0$. In this case $d_0 = 1 - d_1$, then[13] $\phi_0(x) = \frac{1}{2^{l+1}-3}$ when $x \in ((-3 \cdot 2^{l-1}, -2 \cdot 2^{l-1} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \bigcup (-2^{l-1}, 2^{l-1}) \bigcup (2 \cdot 2^{l-1} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 3 \cdot 2^{l-1})) \cap \mathbb{Z}$, otherwise $\phi_0(x) = 0$. Since $\mathbb{E}(b_0) = -\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, we can rewrite b_0 as $b_0 = \langle -\mathbf{s}, \bar{\mathbf{c}} \rangle + e_0$. Finally, we conclude that the result of solving b_0 by the least squares method is $-\mathbf{s}$ as the distribution of e_0 can be approximately regarded as subgaussian (similar to e_1 in Section 3.3) and the variance of e_0 is no more than 2.5 times that of e_1.

In general case, the expectation of b_κ is $(2\kappa - 1)\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$. Then the equation $b_\kappa = [z]_{2^l} \pm d_\kappa \cdot 2^l$ can be turned into $b_\kappa = (2\kappa - 1)\langle \mathbf{s}, \bar{\mathbf{c}} \rangle + e_\kappa$, where e_κ approximately follows subgaussian and its variance is no larger than $(2.5 - 1.5\kappa)\mathbb{D}(e_1)$. As a result, the expectation of the least squares estimator in the new FS-ILWE is $\mathbb{E}(\tilde{\mathbf{s}}) = (2\kappa - 1)\mathbf{s}$. That is, the correction coefficient is $\frac{1}{2\kappa - 1}$.

[13] For brevity, we omit the detailed process of solving the numerical characteristics of random variables.

What remains is how many signatures are needed to recover $\mathbf{s}$. According to Theorem 2, the lower bound of signatures is $m \geq 32\frac{n\tau_e^2}{h}\log(2n)$ where we treat τ_e as the standard deviation of e_κ. In such case, the bound of $\tilde{\mathbf{s}}$ should be transformed to $\|\mathbf{s} - \frac{1}{2\kappa-1}\tilde{\mathbf{s}}\|_\infty < 1/2$ because of the correction coefficient. Hence the number of signatures required in this case is no more than $\frac{\mathbb{D}(e_\kappa)}{\mathbb{D}(e_1)(1-2\kappa)^2} < \frac{5-3\kappa}{2(1-2\kappa)^2}$ times that in the case of $\kappa = 1$.

4 Two Cases of Study: Dilithium and qTESLA

Our randomness leakage attack applies to most known lattice-based Fiat-Shamir signatures. In this section, we show Dilithium and qTESLA are vulnerable.

4.1 Attacks on Dilithium

The Dilithium scheme is built via the "Fiat-Shamir with abort" structure [28,29] and includes several optimizations on top of the Bai-Galbraith scheme [5]. The security of Dilithium is based on the hardness of Module-LWE and Module-SIS problems, a flexible generalization of Ring-LWE and Ring-SIS problems. The signing algorithm is given by Algorithm 4 and we defer the whole description of Dilithium to Appendix D.

Algorithm 2 $\text{Sign}(sk = (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}), \mu \in \mathcal{M})$

1: $\mathbf{A} \sim R_q^{k\times l} \coloneqq \text{Sam}(\rho)$
2: $\mathbf{t}_1 \coloneqq \text{Power2Round}_q(\mathbf{t}, d)$
3: $\mathbf{t}_0 \coloneqq \mathbf{t} - \mathbf{t}_1 \cdot 2^d$
4: $\mathbf{r} \leftarrow \{0,1\}^{256}$
5: $\mathbf{y} \sim S^l_{\gamma_1-1} \coloneqq \text{Sam}(\mathbf{r})$
6: $\mathbf{w} \coloneqq \mathbf{A}\mathbf{y}$
7: $\mathbf{w}_1 \coloneqq \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$
8: $\mathbf{c} \coloneqq \text{H}(\rho, \mathbf{t}_1, \mathbf{w}_1, \mu)$
9: $\mathbf{z} \coloneqq \mathbf{y} + \mathbf{c}\mathbf{s}_1$
10: $(\mathbf{r}_1, \mathbf{r}_0) \coloneqq \text{Decompose}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)$
11: **if** $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}\|_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}_1 \neq \mathbf{w}_1$ **then**
12: goto 4
13: **end if**
14: $\mathbf{h} \coloneqq \text{MakeHint}_q(-\mathbf{c}\mathbf{t}_0, \mathbf{w} - \mathbf{c}\mathbf{s}_2 + \mathbf{c}\mathbf{t}_0, 2\gamma_2)$
15: **if** $\|\mathbf{c}\mathbf{t_0}\|_\infty \geq \gamma_2$ or the number of 1's in $\mathbf{h}$ is greater than ω **then**
16: goto 4
17: **end if**
18: **return** $\sigma = (\mathbf{z}, \mathbf{h}, \mathbf{c})$

In Fiat-Shamir signature schemes, the random oracle used to compute the challenge is implemented by a hash function. We require the entropy of the

challenge is as small as the security parameter. Hence, the challenge set can be seen as a subset of the n-dimension ring R and satisfies the following equation

$$\text{ChSet} = \{c \in R | \|c\|_\infty = 1 \quad \text{and} \quad 2^h \binom{n}{h} \geq 2^\lambda\}$$

where λ is the security parameter. In Dilithium, $n = 256$ and the challenge set consists of 60 non-zero coefficients, denoted as B_{60}.

In Dilithium, although the secret keys consist of $\mathbf{s}_1$ and $\mathbf{s}_2$, the signature $\mathbf{z}$ is only related to $\mathbf{s}_1$ and the proof of knowledge of $\mathbf{s}_2$ is completely removed to significantly decrease the signature size. Therefore, we cannot recover $\mathbf{s}_2$ via our partial randomness leakage attack. Besides, due to the public key compression, we cannot even recover $\mathbf{s}_2$ by the public key $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$. But the work [13,40] showed that just knowing $\mathbf{s}_1$ is sufficient for existential forgery attack.

Obviously, recovering $\mathbf{s}_1$ by leaking the $(l+1)$-th bit of any coefficient of $\mathbf{y}$ is exactly an FS-ILWE problem. For example, in the FS-ILWE problem that is obtained in the key recovery attack on Dilithium-III, $\mathbf{a}$ is the coefficient vector of a 256-degree polynomial with exactly 196 zeros, and the bound $l = 7$ which implies $\|\mathbf{sc}\|_\infty \leq 2^6$ except 1% and the distribution of $\mathbf{e}$ is approximated to subgaussian over a bounded interval $(-2^7, 2^7)$. What remains is to solve some 256-dimension FS-ILWE problems by the least squares method to recover $\mathbf{s}_1$. Note that since Dilithium is a signature based on MLWE, the secret key can be represented by a matrix (for example, a 256×4 matrix in Dilithium-III) and each column of it is an independent vector. In order to recover the secret key, we need to solve 4 independent FS-ILWE problems. Taking advantage of parallel computing, the time needed in this attack is the same as the time needed in a 256-dimensional FS-ILWE problem, however, we need 4 bits of leakages per signature to recover 4 polynomials in the secret key $\mathbf{s}_1$. Experiments on other parameters are performed and the detailed results are described in the experimental section.

4.2 Attacks on qTESLA

Similarly, the qTESLA scheme is also built via the "Fiat-Shamir with aborts" structure and can be seen as a variant of the Bai-Galbraith scheme with a tight security reduction. The main difference between Dilithium and qTESLA is the mathematical structure: Dilithium is based on the hardness of Module-LWE and Module-SIS problems, while qTESLA is based on the hardness of Ring-LWE problem in $\mathbb{Z}_q[x]/(x^n + 1)$. We defer the description of qTESLA to Appendix E.

Compared with Dilithium, our randomness leakage attack can be adapted to qTESLA directly because there is only one polynomial in $\mathbf{s}$ due to the ring structure and we only need one bit per signature to recover the secret key $\mathbf{s}_1$. Moreover, since the public key of qTESLA is not compressed, another component of the signing key $\mathbf{e}$ can be recovered easily after $\mathbf{s}$ is known.

Besides, in an ideal lattice or module lattice based Fiat-Shamir signature scheme, the signature is $\mathbf{z} = \mathbf{y} + \mathbf{sc}$ and $\mathbf{z}, \mathbf{y}, \mathbf{s}, \mathbf{c}$ are all polynomials. That is to say, we can obtain at most n FS-ILWE samples $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ per signature by leaking one bit of n coefficients of $\mathbf{y}$. Obviously, if the required number of

FS-ILWE samples is determined, the number of signatures required for attacking with one bit of leakage is n times of the number with n bits of leakages. Hence, although our attack in Section 3.1 describes how to recover the secret key with only one bit of leakage per signature, for efficiency, we instead leak more than one bit in actual attacks on Dilithium and qTESLA. We will show the number of FS-ILWE samples required in two cases is almost equal in the experimental section.

5 Experimental Results

In the section, we present experimental results of randomness leakage attacks on Dilithium and qTESLA. Specifically, we first describe key recovery attacks on Dilithium and qTESLA by leaking the $(l+1)$-th bit of any coefficient of randomness in Section 5.2, then taking Dilithium as an example, we show how to perform such attack with leakages from other positions in Section 5.3, and with probabilistic leakages in Section 5.4.

5.1 The Leakage Bound l in Dilithium and qTESLA

As discussed in Section 3.1, we need to determine the leakage bound l before attacks. The parameter l is the bound of infinity norm of $\mathbf{sc}$ and is actually given in the parameter sets of Dilithium and qTESLA. However, in order to guarantee that the probability of $\|\mathbf{sc}\|_\infty \geq 2^l$ is negligible, the given bound l is large. In fact, we can lower l as long as it can bound most of $\|\mathbf{sc}\|_\infty$ in our attack. The smaller l is, the less signatures are required to recover the secret key. Hence, we find a suitable l by studying the probability of $\|\mathbf{sc}\|_\infty < 2^l$ on different l statistically for Dilithium and qTESLA.

For Dilithium, Table 1 shows four parameter sets for different security levels. For each set, we randomly choose $10,000$ signatures (corresponding to $2,560,000$ coefficients of $\mathbf{sc}$) and compute the probability of $\|\mathbf{sc}\|_\infty$ within the interval $(-2^l, 2^l)$ when $l = 5, 6, 7, 8, 9$ respectively. The results are displayed in Table 2. In our partial randomness leakage attack, we require that the probability of $\|\mathbf{sc}\|_\infty < 2^{l-1}$ is more than 99%. Based on this, the leakage bound l is set to $8, 8, 7, 7$ for Dilithium-I, Dilithium-II, Dilithium-III and Dilithium-IV.

For qTESLA, the authors specify two parameter sets named qTESLA-p-I and qTESLA-p-III, which are displayed in Table 3. With some minor modifications, the experiments apply to qTESLA. From the statistical results in Table 4, the leakage bound l in qTESLA is set to $8, 9$ and is larger than l in Dilithium.

5.2 Attacking Dilithium and qTESLA

Having determined the leakage bound l, we perform key recovery attacks on Dilithium and qTESLA by leaking the $(l+1)$-th bit of randomness. Our attack consists of three steps: generating signatures with the $(l+1)$-th bit of leakages, then reducing our attack to an FS-ILWE problem and finally solving the

Table 1. Parameters for Dilithium

	description	I weak	II medium	III recommended	IV high
n	dimension	256	256	256	256
q	modulus	8,380,417	8,380,417	8,380,417	8,380,417
h	weight of $\mathbf{c}$	60	60	60	60
γ_1	$\|\mathbf{y}_i\|_\infty \leq \gamma_1 - 1$	523,776 $< 2^{19}$	523,776 $< 2^{19}$	523,776 $< 2^{19}$	523,776 $< 2^{19}$
(k, l)	module parameters	(3, 2)	(4, 3)	(5, 4)	(6, 5)
η	$\|\mathbf{s}_i\|_\infty \leq \eta$	7	6	5	3
β	$\|\mathbf{s}_{1,2}\mathbf{c}\|_\infty \leq \beta$	375	325	275	175
classical security		58	100	138	174
quantum security		53	91	125	158

Table 2. The probability of $\|\mathbf{s}_1\mathbf{c}\|_\infty \leq 2^l$ in Dilithium

	$l = 5$	$l = 6$	$l = 7$	$l = 8$	$l = 9$	leakage bound l
Dilithium-I	0.65127	0.94175	0.99988	1	1	8
Dilithium-II	0.72163	0.97151	0.99999	1	1	8
Dilithium-III	0.80157	0.99078	0.99999	1	1	7
Dilithium-IV	0.95885	0.99997	1	1	1	7

Table 3. Parameters for qTESLA

	description	qTESLA-p-I	qTESLA-p-III
n	dimension	1,024	2,048
q	modulus	343,576,577	856,145,921
h	weight of $\mathbf{c}$	25	40
B	$\|\mathbf{y}_i\|_\infty \leq B$	$2^{19} - 1$	$2^{21} - 1$
σ	sk std. dev.	8.5	8.5
L_S	$\|\mathbf{sc}\|_\infty \leq L_S$	554	901
classical security		150	304
quantum security		139	279

Table 4. The probability of $\|\mathbf{s}_1\mathbf{c}\|_\infty \leq 2^l$ in qTESLA

	$l = 6$	$l = 7$	$l = 8$	$l = 9$	$l = 10$	$l = 11$	leakage bound l
qTESLA-p-I	0.86942	0.99762	1	1	1	1	8
qTESLA–p-III	0.77011	0.984125	0.99999	1	1	1	9

FS-ILWE problem using the least squares method. We run the Dilithium and qTESLA C codes submitted to NIST to obtain signatures and leakage bits in the first step, then use methods presented in Section 3.1 to obtain FS-ILWE samples. Experiments of the first two steps are conducted using C/C++ languages on a single core of an Intel Core(TM) i7-4790 CPU at 3.6GHz. The last step is essentially solving a linear system using the least squares method. Due to the efficient matrix operation in Matlab, we carry out the last step using Matlab R2014b on a desktop with 3.60GHz processor and 12GB memory.

Another point to note is that we leak more than one bit in actual attacks. Taking Dilithium-III as an example, we show the number of FS-ILWE samples required in the case of leaking one bit of randomness per signature and the number in the case of leaking one bit of each coefficient of randomness per signature is almost equal. Fixing sk, we measure the minimum value of m to solve the FS-ILWE problem and the results are displayed in Table 5, which gives the minimum, lower quartile, interquartile mean, upper quartile and maximum numbers of required samples in our 12 trials. As shown in Table 5, the difference of interquartile mean is about 6.32%, meaning that the number of FS-ILWE samples in two cases is not very different. Therefore, to reduce the time of generating signatures, we leak one bit of every coefficient of the randomness polynomial $\mathbf{y}$ in follow-up experiments.

Table 5. Numbers of samples required to recover the secret key of Dilithium-III with different number of leakage bits

leakage bits$^{\#}$	Min	LQ	IQM	UQ	Max
1	1,055,232	1,152,640	1,432,576	1,524,608	1,583,616
256	849,664	1,011,584	1,342,080	1,453,184	1,716,224
DIF	19.48%	12.24%	6.32%	4.68%	-8.37%

Due to the special structure of $\mathbf{c}$, our attack requires a large number of FS-ILWE samples (i.e. signatures). In other words, our attack may run out of memory because we need to solve a linear system with noise consisting of m equations where m is mostly on the order of millions. Some tricks are available to avoid the problem. Since $\mathbf{c}$ is a sparse polynomial with h coefficients ± 1, multiplication by $\mathbf{c}$ can be transformed into an iterated sum over those indices corresponding to the ones. Hence, the complexity of computing $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ is reduced from $O(mn^2)$ and $O(mn)$ to $O(mh^2)$ and $O(mh)$. Moreover, instead of computing $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ directly, we use the block matrix strategy and compute block by block to avoid memory overflow.

Now we turn to experiments on Dilithium and qTESLA. For Dilithium, we perform 12 trails for each set, and our results are displayed in Table 6. Note that n times the given number is the number of FS-ILWE samples or the actual number of signatures required in the case of leaking only one bit per signature.

Not only that, the numbers in Table 6 is the minimum value of m required to recover all coefficients of the secret key polynomial. However, in practice, less signatures are enough since we can recover most of coefficients and then recover the entire secret key by brute force.

Table 6. Numbers of signatures required for attacking Dilithium

	Min	LQ	IQM	UQ	Max
Dilithium-I	10,240	13,191.5	16,066	17,081.5	22,543
Dilithium-II	10,046	11,945.5	14,367.5	16,109.5	17,838
Dilithium-III	3,319	3,951.5	5,242.5	5,676.5	6,704
Dilithium-IV	3,532	3,634.5	3,976	4,127	4,373

Interesting enough, we conclude that the difficulty of our attack is opposite to the difficulty of lattice reduction. The higher the security level is claimed, the less FS-ILWE samples are required and the easier our attack is. The difficulty order of our attack on Dilithium is Dilithium-I > Dilithium-II > Dilithium-III > Dilithium-IV and is consistent with the theoretical results. According to Theorem 2, $m \geq C\frac{n\tau_e^2}{h}\log n$, where n, h are the same for all parameter sets and τ_e is determined by the leakage bound l. When the number of non-zero coefficients of $\mathbf{c}$ is fixed, l is positively related to the value of the secret key $\mathbf{s}$. From Dilithium-I to Dilithium-IV, the secret key is getting smaller and smaller.

Table 7. Average running time for attacking Dilithium

	Time for FS-ILWE samples (ms)	Time for $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ (s)	The total time (s)
Dilithium-I	3.95	17.08	17.084
Dilithium-II	3.51	15.32	15.324
Dilithium-III	1.20	5.54	5.541
Dilithium-IV	0.74	3.50	3.500

In Table 7, we present the running time for our attack on Dilithium. Since the time of generating signatures highly depends on concrete implementations and can be computed given the number of required signatures, we omit it here. Moreover, the running time of solving a linear system consisting of n equations is constant if the dimension n is fixed and negligible[14] and we also omit it. The total time of recovering the secret key of Dilithium is in seconds, making our attack rather practical. The most time-consuming operation is computing $\mathbf{A}^T\mathbf{A}$

[14] Solving a linear system consisting of 256, 1024 and 2048 equations takes about 0.49, 10.5 and 73.2 ms respectively using Matlab.

and $\mathbf{A}^T\mathbf{b}$ in the third step due to the large dimension m, covering 99.97% of the running time.

Similarly, we perform our attack on qTESLA. For each parameter set, we perform 12 trails. Results about the minimum number of required samples and corresponding running time are displayed in Table 8 and 9. The total time for recovering the secret key of qTESLA is within thousands of seconds. We notice that attacks on qTESLA are more difficult than Dilithium, mainly because of the larger dimension n, the sparser polynomial $\mathbf{c}$ and even the larger secret key $\mathbf{s}$. Besides, the sparsity of $\mathbf{c}$ is affected by the dimension n. In general, we can choose a sparser $\mathbf{c}$ when n is larger. Therefore, we may conclude that at the same security level, module lattices are more vulnerable to our attack than ideal lattices because the dimension of its underlying ring is generally smaller. The experimental results also verify that.

Table 8. Numbers of signatures required for attacking qTESLA

	Min	LQ	IQM	UQ	Max
qTESLA-p-I	50,660	84,109.5	108,415.5	152,844	200,463
qTESLA-p-III	94,476	124,692.25	138,652	162,276.75	181,228

Table 9. Average running time for attacking qTESLA

	Time for FS-ILWE samples (ms)	Time for $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ (s)	The total time (s)
qTESLA-p-I	125.64	108.49	108.62
qTESLA-p-III	397.75	1845.05	1845.45

In table 10, we provide numbers of required leakage bits and signatures to attack DSA, ECDSA, Dilithium and qTESLA. It can be seen that lattice-based Fiat-Shamir signatures are easier to attack because less signatures are required when leaking one bit of randomness at almost the same security level. In addition, attacks on DSA and ECDSA take hours, while only a few seconds are required for Dilithium and qTESLA in our attacks.

5.3 Attacking Dilithium by Leaking High-Order Bits

In this section, we perform attacks on Dilithium to show how to recover the secret key with leakage bits of any position between $l+1$ and k of any coefficient of $\mathbf{y}$. For the sake of simplicity, we assume that the leakage positions in all signatures are the same, but our attack applies to the case where signatures leak

Table 10. Numbers of leakage bits and signatures to recover the secret key of (EC)DSA, Dilithium and qTESLA

	Classical security	Leakage bits	Signatures	Work
DSA	160	2	100	[27]
DSA	160	$\log 3 \approx 1.58$	2^{22}	[10]
ECDSA	160	1	2^{33}	[3]
Dilithium	174	1	2^{20}	our work
qTESLA	150	1	2^{27}	our work

at different but known positions. According to Section 5.1, for four parameter sets in Dilithium, the leakage bound l is 7 or 8, hence, we measure the minimum value of m required to recover the secret key for $l = 7, ..., 11$. Here we do not consider larger l, because there is a positive correlation between m and l and larger l requires much more signatures, more memory and longer running time. Experimental results are given in Figure 1.

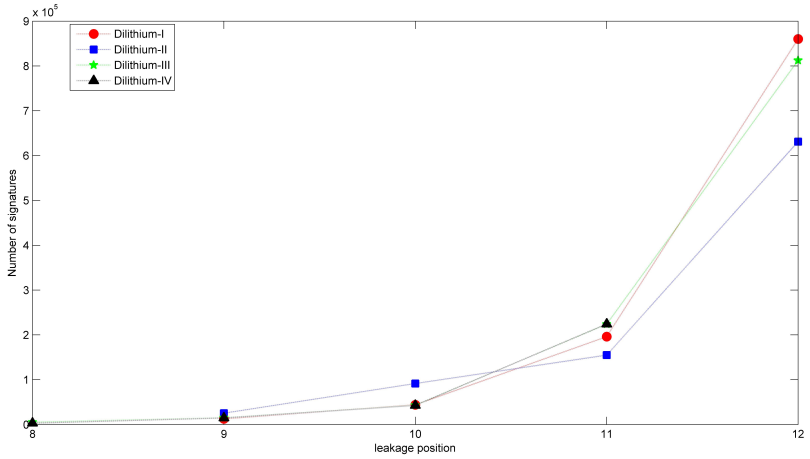

Fig. 1. Number of signatures with high-order bit of leakages

The results in Figure 1 indicate that the value m of required samples for four parameter sets are close when leakage happens at the same position. The results are consistent with the theoretical results. Fixing the dimension n and the number of non-zeros coefficients of $\mathbf{c}$, m is only affected by τ_e, which depends on the leakage bound l. Note that Figure 1 does not contain the result of $l = 11$ for Dilithium-IV because the result of experimental example exceeds $1,000,000$, the maximum value of signatures we set in advance. Another conclusion that

can be drawn from Figure 1 is that when the leakage position is shifted to the left by one, the number of signatures required becomes two to five times. Therefore, for Dilithium where $t = 19$, we conjecture that the number of required signatures with leaking the t-th bit is at most $10^6 \times 5^8 \approx 2^{39}$ (or 2^{47} in the case of leaking only one bit per signature), which is less than 2^{64}, the maximum number of signatures that adversaries can obtain by NIST [33]. In other words, our attack is theoretically applicable to the case of leaking the highest order bit of $\mathbf{y}$, however, it is not feasible in practice due to memory and time limits.

5.4 Attacking Dilithium with Probabilistic Bits

Section 3.4 shows that, our key recovery attack against lattice-based Fiat-Shamir signatures still holds as long as the probability that an adversary obtains the randomness bit is larger than 0.5. In this section, we validate our method by attacking Dilithium-IV using randomness bit with different probability, since Dilithium-IV is the easiest to attack as shown in Table 6. Besides, for simplicity, we assume that leakage occurs at the $(l+1)$-th bit (i.e. the 8-th bit for Dilithium-IV).

Table 11. Numbers of signatures required for attacking Dilithium-IV using probabilistic randomness bits

probability	Min	LQ	IQM	UQ	Max
1	3,532	3,634.5	3,976	4,127	4,373
0.9	4,300	5,512	6,576	7,878.5	9,011
0.8	8,659	11,740	13,943.5	15,942.5	17,323
0.7	23,980	27,238	30,084.5	34,933.5	46,931
0.6	107,797	124,579	142,574.5	162,517	168,280

For one bit, the leakage probability 0.5 can be regarded as no leakages since it is equivalent to random guessing, thus 0.6 is a fairly low and practical probability. Hence, we set the leakage probability to be $1, 0.9, 0.8, 0.7, 0.6$. For each probability, we perform 12 trails. The minimum value of required signatures to recover the secret key is displayed in Table 11. For example, even if the leakage probability is as low as 0.6, less than two hundred thousand signatures are sufficient to recover the secret key of Dilithium-IV, which is almost 40 times the number of signatures required when the leakage probability is 1.

6 A Proof-of-Concept Practical Experiment

In previous sections, we focus on mathematical techniques to exploit randomness leakage to recover the secret key. In this section, we perform a non-profiled power analysis attack to show how to come up with the required bit of randomness in practice.

6.1 Available Randomness Bits in Lattice-based Fiat-Shamir Signatures

As said in Section 3.1, the randomness y is much larger than $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, and the signature z is the sum of a large number y and a small number $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$. Denote the $(k-l)$ MSBs of randomness y by $\mathrm{MSB}_l(y) = (y - [y]_{2^l})/2^l$ where k is the length of y, and $\mathrm{MSB}_l(z) = (z - [z]_{2^l})/2^l$. Suppose that $\Pr[|\langle \mathbf{s}, \bar{\mathbf{c}} \rangle| < 2^{l-1}] \approx 1$. Thus, we have $\mathrm{MSB}_l(y) = \mathrm{MSB}_l(z)$ if no overflow occurs when computing $[y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, otherwise $\mathrm{MSB}_l(y) = \mathrm{MSB}_l(z) + 1$. That is, the high-order bits of some randomness of y are available, which are revealed by the signature z. With this interesting observation, we may wonder: which randomness bits are available? And can we exploit these available randomness bits to break the signature schemes?

For the former, we note that if no overflow occurs when computing $[y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, or even if there is an overflow, but we can decide how many locations are affected, those high-order randomness bits of unaffected locations are known according to the signature z. Take $z > 0$ as an example. For the $(l+1)$-th bit of y (denoted by y_{l+1}), since carry may occur if $z_l = 0$ and borrow may occur if $z_l = 1$, we cannot infer y_{l+1} from z_{l+1}. However, for y_{l+2}, if $z_{l,l+1} \neq 00$ or 11, there will definitely be no overflow and $y_{l+2} = z_{l+2}$. And so on, for y_{l+c} where $c \geq 2$, if $z_{l,\ldots,l+c-1} \neq 0 \ldots 0$ or $1 \ldots 1$, we have $y_{l+c} = z_{l+c}$. Similar conclusion holds for $z < 0$. Hence, $\Pr[y_{l+c} \text{ is available}] = 1 - \frac{1}{2^{c-1}}$. Note that $\Pr[y_{l+c} \text{ is available}] \neq \Pr[y_{l+c} = z_{l+c}]$.

Now we turn to the second question: can these available randomness bits be exploited? From the theoretical perspective, available randomness bits are determined according to the signature, and the adversary cannot obtain additional information other than the signature from available randomness. Hence, the theoretical security of a provably secure signature scheme is unaffected. To be clear here, although it seems that available randomness bit is exactly the required high-order randomness bit in our attack, only these available randomness bits come from signatures are not enough for our attack, since rejection sampling technique fundamentally eliminates such statistical attacks. In other words, those randomness bits cannot be inferred from z are critical for our attack, and that is why we can bypass the rejection sampling technique to recover the secret key.

However, from the practical perspective, available randomness may affect the practical security of lattice-based Fiat-Shamir signatures. First, suppose M leakage bits are required when leaking the $(l+c)$-th bit of y ($c \geq 2$) in our attack, now only $\frac{1}{2^{c-1}}$ of M bits need to leak by SCAs, and the rest are available. More importantly, available randomness can be exploited in practical leakage attacks to obtain information of randomness. In the following part, we show how to recover the required randomness bit by a non-profiled attack with the help of available randomness bits.

6.2 A Non-Profiled Power Attack on Polynomial Addition

Our attack is described in Figure 2, in which I denotes the sensitive data in target device and L denotes the collected power traces from this device. The

adversary sends a number of messages msg to the target device (step ①), which sends back the corresponding signature sig but also leaks some information L physically (step ②). In the analysis step ③, these power traces can be divided into two groups through the signature sig (which has been shown in 6.1): in one group, the sensitive data (high-order bits of the randomness y) are available, and the sensitive data and corresponding power traces of this group are denoted by $I_{available}$ and $L_{available}$ respectively. However, the sensitive data of the other group are still uncertain, these data and their power traces are denoted by $I_{uncertain}$ and $L_{uncertain}$. Next, the adversary is able to build a profiled model T for each high-order bit of y using $I_{available}$ and $L_{available}$. Thus, $I_{uncertain}$ can be recovered utilizing the obtained model T and power traces $L_{uncertain}$. Combining $I_{available}$ and $I_{uncertain}$, the adversary could obtain all high-order bits of randomness y.

Although the profiling is utilized, we believe that our attack belongs to non-profiled attacks due to the way of obtaining the profiled data and the time to profile: (1) **How to obtain the profiled data.** Despite the profiling is mounted in our attack, the sensitive data used to profile is obtained through public signatures, and the profiling device is not needed in our attack. (2) **When to profile.** The profiling is mounted in step ③, which implies the profiling phase is a step in the analysis. Namely, our attack just makes use of the idea and procedure of profiled attacks, but the power of the adversary is totally the same with that in non-profiled attacks.

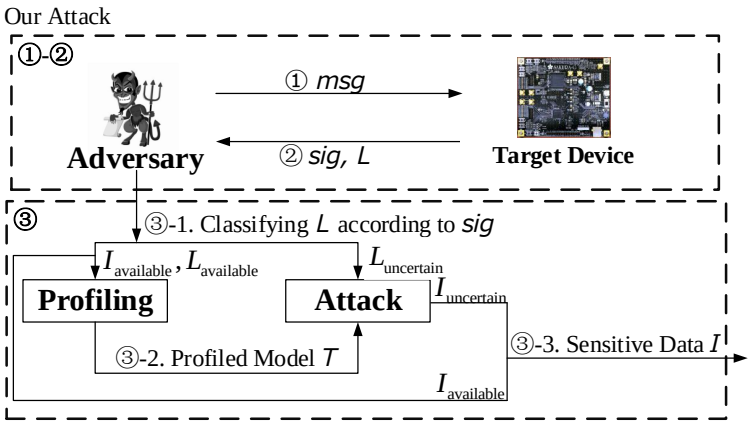

Fig. 2. Overview of Proof-of-the-Concept Real-World Attack (which also explains why it belongs to non-profiled attacks).

This attack has a main advantage: **Powerful**. It belongs to non-profiled SCAs, but its idea and procedure are similar with profiled attacks. Thus, it can be much more powerful than other non-profiled attacks.

The target process of our attack is the addition operation $\mathbf{z} = \mathbf{y} + \mathbf{sc}$ in the signing algorithm of Dilithium. In the Dilithium software, the addition $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ is implemented as $z + 2q = (y + q) + (\langle \mathbf{s}, \bar{\mathbf{c}} \rangle + q)$. Although the magnitude of $y + q$ and $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle + q$ is the same, high-order bits of some $y + q$ still can be inferred from $z + q$. Here the MSBs of $y + q$ and $z + q$ are the same if no overflow occurs when computing $[y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle + [q]_{2^l}$. In this section, we take Dilithium-II (where $l = 8$) as an example to show how to obtain 1 bit information of $y + q$ through power traces and there are available randomness bits when the leakage position $t \geq 10$. Note that in order to be consistent with previous sections, we still use y and z instead of $y + q$ and $z + q$ below.

A software platform used to implement the sensitive addition operation related to y is a microcontroller (MCU) 8051 STC89C58RD+ clocked at 11.0592MHz, and the power traces of the device are measured by a oscilloscope (Agilent DSO9104A). The sampling rate is set to 20MSa/s. We measured the voltage drop over a 50Ω resistor in the GND path of MCU as the power consumption. For one trace, there are 65,000 samples, which are around the sensitive operation. Totally 10,000 traces are collected with a lower pass filter (BLP-90+).

We take our attack on the 10-th bit of y as an example. First, we analyze the public signature z to find if the 10-th bit of z equals to that of y. Actually, we get 5,171 satisfied traces, then we use the T-test to detect the leakages with these traces, which is shown as Fig 3.

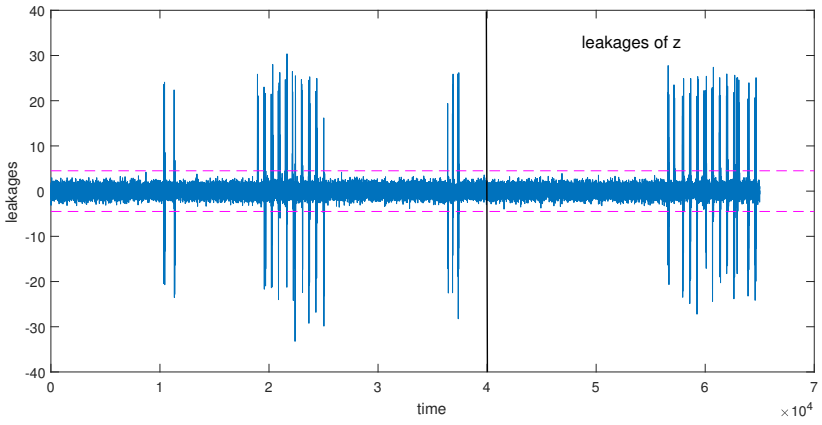

Fig. 3. The leakages of the 10-th bit using public signature z.

In fact, leakage detection can filter out Points of Interesting (PoIs) based on 10-th bit of z. Since 10-th bit of z definitely equals to that of y, the detected points are related to both y and z. To profile a premise template for y, it is necessary to remove the PoIs only related to z. In fact, y is the input of this addition operation while z is the output, and we regard the PoIs after $40{,}000^{th}$

(black line in Fig 3) as leakages of outputs (i.e. z). While profiling and attacking, we only use first 40,000 points in each trace. We profile the template with 5,171 satisfied traces, and use the obtained template to attack with the other 4,829 traces. The results show that the 10-th bit of y can be recovered with 99.63% success rate. Other bits of y can be recovered similarly, and the results are shown in Table 12.

Table 12. The results aiming to recover high-order bit in y.

i-th bit	trace$^{\#}$ for profiling	trace$^{\#}$ for attacking	PoI$^{\#}$	Success Rate
10	5171	4829	134	**99.63%**
11	7654	2346	105	**100%**
12	8854	1146	283	**100%**
13	9444	556	306	**100%**
14	9727	273	320	**100%**
15	9865	135	367	**100%**
16	9927	73	605	**100%**
17	9959	41	530	**100%**
18	9974	26	537	**100%**
19	9988	12	581	**100%**

It is obvious that our attack achieves perfect success rates under our experimental setting. However, this experiment is too far from a practical validation.

6.3 Attacking under Artificial Noisy Setting

To make a more convincing validation of our attacking, we simulate the practical environment by adding artificial Gaussian noise to existing traces then evaluate our attack again. All attacking parameters (such as number of traces for profiling and attacking, number of PoIs) are the same as those in Section 6.2. We use σ to denote the standard deviation of Gaussian noise added. The results are shown in Fig. 4.

It should be noted that, when σ is up to 10, the highest SNR of POIs among these traces is 0.1938, which is relatively practical (the highest SNR of PoIs in ASCAD dataset [39] is over 12). The results in Fig. 4 show that Success Rates of our attack for all target bits are higher than 0.65. We have shown how to recover the secret key under an assumption, which claims the adversary can get high-order bits of randomness y with a leakage probability over 0.6. Hence this additional experiment validates this assumption should be reasonable in practice.

In this work, we just provide a method to exploit available randomness, but we conjecture there must be other applications, making it easier to leak

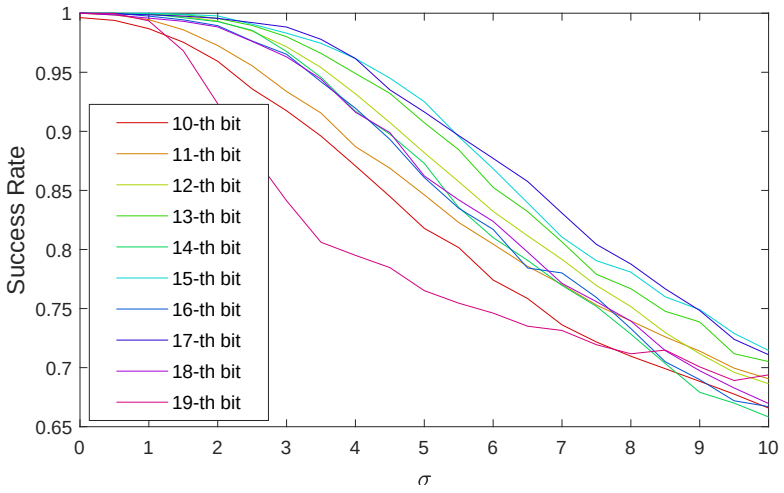

Fig. 4. The results of our attack after adding artificial Gaussian noise to existing traces.

randomness of lattice-based Fiat-Shamir signatures, and thus much easier to recover the secret key.

7 Conclusion

In this work we presented a polynomial time attack on lattice-based Fiat-Shamir signatures with only one randomness bit of leakage per signature. We showed that the key recovery attack with randomness leakage can be reduced to the FS-ILWE problem, which can be solved efficiently by the least squares method. Our attack was experimentally validated to recover secret keys of Dilithium and qTESLA. In addition, we showed how to obtain the required leakage bits in practice using a non-profiled attack.

7.1 Future Work

The leakage in our key recovery attack may occur at any position of randomness except the l LSBs where l is the leakage bound satisfying the condition $\Pr[\|\mathbf{sc}\|_\infty < 2^{l-1}] \geq 99\%$. Namely, our attack fails with low-order leakage bits. Although we believe that leakages of low-order bits of randomness reveal information of the secret key, attacks exploiting these bits are still an open issue.

Note that our key recovery attack is applicable to most of lattice-based Fiat-Shamir signatures except BLISS [17]. To improve the success rate of the rejection sampling, BLISS uses a bimodal Gaussian distribution and the signature is $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{sc}$ where $b \in \{0, 1\}$ is kept hidden. As a result, the linear system in the last step of our attack contains $\mathbf{s}$ and $-\mathbf{s}$ which will cancel each other out.

We expect to extend our attack to make it applicable to all the Fiat-Shamir signatures over lattice including BLISS in the future.

7.2 Discussion of Possible Countermeasures

To protect against our attacks, we provide some rough ideas here. Ultimately, both of our attacks based on leakages of randomness used in a realistic signature scheme, and side-channel leakage is the most important and common one. Thus, we discuss the possible countermeasures borrowed from the side-channel literature.

Since any step (such as sampling and computation) that involves the manipulation of randomness in the signing algorithm may leak the information of randomness, a generic countermeasure against side-channel attacks is masking the whole signing algorithm. In 2018, Barthe et al. [6] proposed the first arbitrary-order masking of the GLP signature scheme [21], which can be seen as the ancestor of Dilithium and qTESLA. Later, this work led to the masking schemes of Dilithium [32] and qTESLA [22]. In theory, the masked signatures are secure against side-channel attacks. However, randomness of above three signatures follow the uniform distribution. As shown in Section 1.3, the distribution of lattice-based Fiat-Shamir signatures can also be the Gaussian distribution. In 2019, a masking scheme for be the Gaussian distribution was proposed by Barthe et al. [7] as well.

In addition, for signatures using the Gaussian distribution, the sampling procedure is also a potential side-channel leakage source. For such attacks, one possible countermeasure is implementing the Gaussian sampling in constant time, such as the work [31]. Alternatively, using the uniform sampler instead of the Gaussian sampler to generate randomness. It should be emphasized that these countermeasures are only for variable-time Gaussian sampling and cannot prevent other potential leakage sources.

References

1. Onur Acıiçmez, Billy Bob Brumley, and Philipp Grabher. New results on instruction cache attacks. In *CHES*, pages 110–124, 2010.
2. Thomas Allan, Billy Bob Brumley, Katrina Falkner, Joop van de Pol, and Yuval Yarom. Amplifying side channels through performance degradation. In *CCS*, pages 422–435. ACM, 2016.
3. 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 *ASIACRYPT*, pages 262–281, 2014.
4. Lszl Babai. On lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
5. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, pages 28–47, 2014.
6. Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi. Masking the GLP lattice-based signature scheme at any order. In *EUROCRYPT*, pages 354–384, 2018.

7. Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi, and Mehdi Tibouchi. GALACTICS: gaussian sampling for lattice-based constant-time implementation of cryptographic signatures, revisited. In *CCS*, pages 2147–2164. ACM, 2019.
8. Naomi Benger, Joop van de Pol, Nigel P. Smart, and Yuval Yarom. "ooh aah... just a little bit" : A small amount of side channel can go a long way. In *CHES*, pages 75–92, 2014.
9. Nina Bindel, Sedat Akleylek, Erdem Alkim, Paulo S. L. M. Barreto, Johannes Buchmann, Edward Eaton, Gus Gutoski, Juliane Krämer, Patrick Longa, Harun Polat, Jefferson E. Ricardini, and Gustavo Zanon. qTESLA. Submission to the NIST Post-Quantum Cryptography Standardization, 2017. `https://tesla.informatik.tu-darmstadt.de/de/tesla/`.
10. Daniel Bleichenbacher. On the generation of DSS one-time keys. Manuscript. The result was presented at the Monteverita workshop in March 2001.
11. Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *CRYPTO*, pages 129–142, 1996.
12. Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. In *ASIACRYPT*, pages 494–524, 2018.
13. Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *CHES*, 2018(3):21–43, 2018.
14. Billy Bob Brumley and Risto M. Hakala. Cache-timing template attacks. In *ASIACRYPT*, pages 667–684, 2009.
15. Billy Bob Brumley and Nicola Tuveri. Remote timing attacks are still practical. In *ESORICS*, pages 355–371, 2011.
16. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.
17. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO*, pages 40–56, 2013.
18. 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 *CCS*, pages 1857–1874. ACM, 2017.
19. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1987.
20. Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, gauss, and reload – a cache attack on the BLISS lattice-based signature scheme. In *CHES*, pages 323–345, 2016.
21. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, pages 530–547, 2012.
22. Franois Grard and Mlissa Rossi. An efficient and provable masked implementation of qTESLA. In *CARDIS*, pages 74–91, 2019.
23. Nadia Heninger and Hovav Shacham. Reconstructing RSA private keys from random key bits. In *CRYPTO*, pages 1–17, 2009.
24. Nick A Howgrave-Graham and Nigel P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, 2001.
25. Daniel Hsu, Sham M Kakade, and Tong Zhang. Tail inequalities for sums of random matrices that depend on the intrinsic dimension. *Electronic Communications in Probability*, 17(14):1–13, 2012.

26. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

27. Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In *CT-RSA*, pages 293–309, 2013.

28. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.

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

30. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, and Stehlé Damien. CRYSTALS-Dilithium. Submission to the NIST Post-Quantum Cryptography Standardization, 2017. `https://pq-crystals.org/dilithium`.

31. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In *CRYPTO*, pages 455–485, 2017.

32. Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking Dilithium. In *ACNS*, pages 344–362, 2019.

33. National Institute of Standards on Technology (NIST). Post-quantum cryptography standardization. `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization`.

34. National Institute of Standards on Technology (NIST). Status report on the second round of the nist post-quantum cryptography standardization process. `https://csrc.nist.gov/publications/detail/nistir/8309/final`.

35. Nguyen and Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15(3):151–176, 2002.

36. Phong Q. Nguyen and Igor E. Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Designs, Codes and Cryptography*, 30(2):201–217, 2003.

37. Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To BLISS-B or not to be: Attacking strongswan's implementation of post-quantum signatures. In *CCS*, pages 1843–1855. ACM, 2017.

38. Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In *CHES*, pages 513–533, 2017.

39. Emmanuel Prouff, Rémi Strullu, Ryad Benadjila, Eleonora Cagli, and Cécile Dumas. Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. Cryptology ePrint Archive, Report 2018/053, 2018. `http://eprint.iacr.org/2018/053`.

40. Prasanna Ravi, Mahabir Prasad Jhanwar, James Howe, Anupam Chattopadhyay, and Shivam Bhasin. Side-channel assisted existential forgery attack on Dilithium - a NIST PQC candidate. Cryptology ePrint Archive, Report 2018/821, 2018. `https://eprint.iacr.org/2018/821`.

41. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1990.

42. Joop van de Pol, Nigel P. Smart, and Yuval Yarom. Just a little bit more. In *CT-RSA*, pages 3–21, 2015.

A Proofs of Lemma 4 and Lemma 5

A.1 Proof of Lemma 4

Since X_i's are independent τ_i-subgaussian variables, for all $s \in \mathbb{R}$, we have:

$$\begin{aligned}\mathbb{E}[\exp(sX)] &= \mathbb{E}[\exp(s(\mu_1 X_1 + \cdots + \mu_n X_n))] \\ &= \mathbb{E}[\exp(\mu_1 s X_1) \ldots \exp(\mu_n s X_n)] = \prod_{i=1}^{n} \exp(\mu_i s X_i) \\ &\leq \prod_{i=1}^{n} \exp(\frac{s^2(\mu_i \tau_i)^2}{2}) = \exp(\frac{s^2 \tau^2}{2})\end{aligned}$$

with $\tau_2 = \mu_1^2 \tau_1^2 + \cdots + \mu_n^2 \tau_n^2$ are required.

A.2 Proof of Lemma 5

Fix a unit vector $\mathbf{u}_0 \in \mathbb{R}^m$,

$$\langle \mathbf{u}_0, \mathbf{y} \rangle = \langle \mathbf{A}^T \mathbf{u}_0, \mathbf{x} \rangle = \mu \langle \mathbf{u}, \mathbf{x} \rangle$$

where $\mu = ||\mathbf{A}^T \mathbf{u}_0||_2$, and $\mathbf{u} = \frac{1}{\mu} \mathbf{A}^T \mathbf{u}_0$ is a unit vector of $\mathbb{R}^n$. Since $\mathbf{x}$ is τ-subgaussian, the inner product $\langle \mathbf{u}, \mathbf{x} \rangle$ is a τ-subgaussian variable. As a result, $\langle \mathbf{u}_0, \mathbf{y} \rangle = \mu \langle \mathbf{u}, \mathbf{x} \rangle$ is $(|\mu|\tau)$-subgaussian by Lemma 4. However, by definition of the operator norm, $|\mu| \leq \|\mathbf{A}^T\|_2^{\text{op}}$, and the result follows.

B Attack without Leakage

Up to now, we can recover the secret key of lattice-based Fiat-Shamir signatures with only one bit leakage of the randomness used in the signing algorithm per signature. A natural question we may wonder is that whether such an attack is still applicable without leakage. Or equivalently, can we recover the secret key only with signatures? Unfortunately, the answer is no and detailed explanations are given from two respects blow.

As we mentioned already, the leaked $(l+1)$-th bit of y is essential to remove the modulus in (3) and in the absence of leakage, the attack can be reduced to another LWE variant:

$$[z]_{2^l} = [y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle \pmod{q} \tag{12}$$

where we let

$$\mathbf{a} = \bar{\mathbf{c}} \quad \text{and} \quad e = [y]_{2^l} \quad \text{and} \quad b = [z]_{2^l} \quad \text{and} \quad q = 2^l.$$

This type of LWE has the following properties:

– The modulus q is very small (only l bits and $l = 7$ or 8 for Dilithium);
– The error e has the same magnitude as the modulus q ($e \in (-q, q)$);
– The dimension n may be small ($n = 256$ for Dilithium).

We will give a formal definition of the LWE variant with large errors (LLWE) and show it is hard in the information-theoretic sense.

Definition 7 (LLWE). *For any vector $\mathbf{s} \in \mathbb{Z}^n$, the LLWE distribution over $\mathbb{Z}^n \times \mathbb{Z}_p$ are of the form*

$$(\mathbf{a}, b) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \pmod q)$$

where $\mathbf{a} \leftarrow B_h, e \leftarrow \chi_e^{(\mathbf{a},\mathbf{s})}$ such that $|\langle \mathbf{a}, \mathbf{s} \rangle| < q$.

It is obvious that given two vectors $\mathbf{s}, \mathbf{s}' \in \mathbb{Z}^n$, the LLWE distributions $\mathcal{D}_{\mathbf{s}}, \mathcal{D}_{\mathbf{s}'}$ are the same if nothing is known about e, namely, it it impossible for adversaries to distinguish $\mathcal{D}_{\mathbf{s}}, \mathcal{D}_{\mathbf{s}'}$. Hence, the LLWE problem is hard in the information-theoretic sense and our attack cannot extend to the leak-free setting.

Step back, another possible attack is reducing the whole signature to the FS-ILWE problem:

$$z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle \tag{13}$$

where we let

$$\mathbf{a} = \bar{\mathbf{c}} \quad \text{and} \quad e = y \quad \text{and} \quad b = z.$$

Since Fiat-Shamir signature over lattice is computed without modular reduction, the signature is exactly an FS-ILWE problem. However, we cannot recover the secret key by solving such an FS-ILWE problem because signatures $\mathbf{z}$ are filtered by the rejection sampling, which provides that $\mathbf{z}$ are independent from the secret key $\mathbf{s}$. Therefore, to some extent, the rejection sampling techniques fundamentally eliminates the potential threat of statistical attacks like ours in the leak-free setting.

In general, lattice-based Fiat-Shamir signatures are secure against attacks using statistical approaches and leakage of randomness is the necessary condition to recover the secret key for this type attacks.

C Remove the Heuristic Assumptions

Totally speaking, our proof is established on two heuristic assumptions: the first is assuming we can always guess carry or borrow correctly and the second is treating the error term as subgaussian approximately. The first is easy to remove, because we can simply find l so that $\Pr[\|\mathbf{sc}\|_\infty \leq 2^{l-1}] \approx 100\%$ except negligible probability, which is not hard to satisfy.

The idea to remove the second assumption based on the fact that the expectation of the error term is $\mathbb{E}([y]_{2^l}) = -\frac{2^l-1}{2^{\gamma+1}-1}\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, which is very small and is

proportional to $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$. Let $e' = e - \mathbb{E}([y]_{2^l})$, then $\mathbb{E}(e') = 0$ and e' is subgaussian obviously. Rewrite the signature as

$$b = [z]_{2^l} = \langle \mathbf{s}, \bar{\mathbf{c}} \rangle + e = \langle \mathbf{s}, \bar{\mathbf{c}} \rangle - \frac{2^l - 1}{2^{\gamma+1} - 1} \langle \mathbf{s}, \bar{\mathbf{c}} \rangle + e + \frac{2^l - 1}{2^{\gamma+1} - 1} \langle \mathbf{s}, \bar{\mathbf{c}} \rangle = \langle (1 - \frac{2^l - 1}{2^{\gamma+1} - 1})\mathbf{s}, \bar{c} \rangle + e' \tag{14}$$

then $\mathbf{b} = \mathbf{C}\mathbf{s} + \mathbf{e}$ is equivalent to the form $\mathbf{b} = \mathbf{C}(1 - \frac{2^l-1}{2^{\gamma+1}-1})\mathbf{s} + \mathbf{e}'$, where the coefficient e' of $\mathbf{e}'$ satisfies the equation $e' = e + \frac{2^l-1}{2^{\gamma+1}-1}\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ thus subgaussian.

Let $\mathbf{s}' = (1 - \frac{2^l-1}{2^{\gamma+1}-1})\mathbf{s}$, thus the problem $\mathbf{b} = \mathbf{C}\mathbf{s}' + \mathbf{e}'$ is an FS-ILWE problem whose error term distribution is subgaussian and can be solved by the least squares method as shown in Section 3.3. If we can establish the bound $\|\mathbf{s}' - \tilde{\mathbf{s}}\|_\infty < 1/2 - \frac{2^l-1}{2^{\gamma+1}-1}\|\mathbf{s}\|_\infty$, then we can recover the secret key $\mathbf{s} = \lfloor \tilde{\mathbf{s}} \rceil$. The bound can be easily obtained by applying Lemma 6 with $t' = 1/2 - \frac{2^l-1}{2^{\gamma+1}-1}\|\mathbf{s}\|_\infty$ instead of $t = 1/2$ in Theorem 2. According to Lemma 6 and Theorem 1, if we need m samples to recover $\mathbf{s}$ that satisfies $\|\tilde{\mathbf{s}} - \mathbf{s}\|_\infty < 1/2$ with probability at least $1 - \frac{1}{2n} - 2^{-\eta}$, then $m' = (\frac{t}{t'})^2 m$ samples are enough to ensure $\|\tilde{\mathbf{s}} - \mathbf{s}'\|_\infty < 1/2 - \frac{2^l-1}{2^{\gamma+1}-1}\|\mathbf{s}\|_\infty$ with the same probability.

In general, since $\frac{2^l-1}{2^{\gamma+1}-1}\|\mathbf{s}\|_\infty$ is too small [15], the FS-ILWE whose error term distribution is $\chi_e^{(\mathbf{a},\mathbf{s})}$ can be solved by reducing it to an FS-ILWE with subgaussian, in which we can recover $\mathbf{s}' = (1 - \frac{2^l-1}{2^{\gamma+1}-1})\mathbf{s}$.

D Description of Dilithium

The Dilithium scheme is built via the "Fiat-Shamir with abort" structure [28,29] and includes several optimizations on top of the Bai-Galbraith scheme [5]. The security of Dilithium is based on the hardness of Module-LWE and Module-SIS problems, a flexible generalization of Ring-LWE and Ring-SIS problems. The Dilithium scheme is given by Algorithms 3-5.

The secret keys $\mathbf{s}_1, \mathbf{s}_2$ are generated by an extendable output function Sam, a function on bit strings whose output can be extended to any desired length, and have uniformly random coefficients in the range $[-\eta, \eta]$. The Power2Round$_q$ algorithm is used to partition each coefficient of the MLWE instance $\mathbf{t}$ into high-order bits and low-order bits respectively. The public key includes a seed ρ used to compute the matrix $\mathbf{A}$ by Sam and $\mathbf{t}_1$ associated to the $\lceil \log q \rceil - d$ high-order bits of $\mathbf{t}$.

To sign a message μ, the signer firstly computes the randomness vector $\mathbf{y}$ using the Sam algorithm, then computes the challenge $\mathbf{c}$ and finally computes the signature candidate $\mathbf{z}$. If all the checks in Line 11 and 15 pass, output the signature $\mathbf{z}$, otherwise the signing algorithm restarts until a signature is valid. Since the public key is compressed, the signer needs to provide a "hint" for the

[15] For example, in Dilithium with the recommended parameters, $\|\mathbf{s}\|_\infty \leq 5$ and $\frac{2^l-1}{2^{\gamma+1}-1}\|\mathbf{s}\|_\infty \leq 0.0006$. Moreover, the number of required samples computing $\mathbf{s}'$ is 0.24% more than that required computing $\mathbf{s}$ in Section 3.3.

verifier to compute the challenge in the verification algorithm. The algorithm MakeHint_q is used to make such a hint and the algorithm UseHint_q in the verifying algorithm shows how to use the hint to complete the verification. For completeness, we also describe the verification algorithm in Algorithm 5.

Algorithm 3 KeyGen()

$\rho, \rho' \leftarrow \{0,1\}^{256}$
$\mathbf{A} \sim R_q^{k \times l} \coloneqq \text{Sam}(\rho)$
$(\mathbf{s}_1, \mathbf{s}_2 \sim S_\eta^l \times S_\eta^k \coloneqq \text{Sam}(\rho')$
$\mathbf{t} \coloneqq \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
$\mathbf{t}_1 \coloneqq \text{Power2Round}_q(\mathbf{t}, d)$
return $(vk = (\rho, \mathbf{t}_1), sk = (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}))$

Algorithm 4 $\text{Sign}(sk = (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}), \mu \in \mathcal{M})$

$\mathbf{A} \sim R_q^{k \times l} \coloneqq \text{Sam}(\rho)$
$\mathbf{t}_1 \coloneqq \text{Power2Round}_q(\mathbf{t}, d)$
$\mathbf{t}_0 \coloneqq \mathbf{t} - \mathbf{t}_1 \cdot 2^d$
$\mathbf{r} \leftarrow \{0,1\}^{256}$
$\mathbf{y} \sim S_{\gamma_1 - 1}^l \coloneqq \text{Sam}(\mathbf{r})$
$\mathbf{w} \coloneqq \mathbf{A}\mathbf{y}$
$\mathbf{w}_1 \coloneqq \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$
$\mathbf{c} \coloneqq \text{H}(\rho, \mathbf{t}_1, \mathbf{w}_1, \mu)$
$\mathbf{z} \coloneqq \mathbf{y} + \mathbf{c}\mathbf{s}_1$
$(\mathbf{r}_1, \mathbf{r}_0) \coloneqq \text{Decompose}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)$
if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}\|_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}_1 \neq \mathbf{w}_1$ **then**
    goto 4
end if
$\mathbf{h} \coloneqq \text{MakeHint}_q(-\mathbf{c}\mathbf{t}_0, \mathbf{w} - \mathbf{c}\mathbf{s}_2 + \mathbf{c}\mathbf{t}_0, 2\gamma_2)$
if $\|\mathbf{c}\mathbf{t}_\mathbf{0}\|_\infty \geq \gamma_2$ or the number of 1's in $\mathbf{h}$ is greater than ω **then**
    goto 4
end if
return $\sigma = (\mathbf{z}, \mathbf{h}, \mathbf{c})$

E Description of qTESLA

Similarly, the qTESLA scheme is also built via the "Fiat-Shamir with aborts" structure and can be seen as a variant of the Bai-Galbraith scheme with a tight security reduction. The main difference between Dilithium and qTESLA is the mathematical structure: Dilithium is based on the hardness of Module-LWE and Module-SIS problems, while qTESLA is based on the hardness of Ring-LWE problem in $\mathbb{Z}_q[x]/(x^n + 1)$. The simplified qTESLA scheme is given by Algorithm 6-8[16].

[16] The qTESLA scheme submitted to NIST is deterministic and for simplicity here we present the non-deterministic version with some minor modifications.

Algorithm 5 Verify($vk = (\rho, \mathbf{t}_1), \mu \in \mathcal{M}, \sigma = (\mathbf{z}, \mathbf{h}, c)$)

$\mathbf{A} \sim R_q^{k \times l} \coloneqq \text{Sam}(\rho)$
$\mathbf{w}_1 \coloneqq \text{UseHint}_q(\mathbf{h}, \mathbf{Az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2)$
if $\mathbf{c} = \text{H}(\rho, \mathbf{t}_1, \mathbf{w}_1, \mu)$ and $\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta$ and the number of 1's in $\mathbf{h}$ is less than ω **then**
    return 1
else
    return 0
end if

Algorithm 6 KeyGen()

$\text{seed}_a \leftarrow \{0,1\}^{256}$
$\mathbf{a} \sim R_q \coloneqq \text{GenA}(\text{seed}_a)$
while $\mathbf{s}$ and $\mathbf{e}$ do not fulfill certain criteria **do**
    $\mathbf{s} \sim R_q \leftarrow D_\sigma, \mathbf{e} \sim R_q \leftarrow D_\sigma$
end while
$\mathbf{t} = \mathbf{as} + \mathbf{e} \bmod q$
return $(vk = (\text{seed}_a, \mathbf{t}), sk = (\mathbf{s}, \mathbf{e}, \text{seed}_a)$

Algorithm 7 Sign($sk = (\mathbf{s}, \mathbf{e}, \text{seed}_s), \mu \in \mathcal{M}$)

$\mathbf{a} \sim R_q \coloneqq \text{GenA}(\text{seed}_a)$
while Reject($\mathbf{z}$, $\mathbf{v}$, c, $\mathbf{s}$) **do**
    $\text{seed}_y \leftarrow \{0,1\}^{256}$
    $\mathbf{y} \sim R_q \coloneqq \text{GenY}(\text{seed}_y)$
    $\mathbf{v} \coloneqq \mathbf{ay} \bmod q$
    $\mathbf{c} \coloneqq \text{H}(\text{Round}(\mathbf{v}), \mu)$
    $\mathbf{z} = \mathbf{y} + \mathbf{sc}$
end while
return $\sigma = (\mathbf{z}, \mathbf{c})$

Algorithm 8 Verify($vk = (\text{seed}_a, \mathbf{t}), \mu \in \mathcal{M}, \sigma = (\mathbf{z}, \mathbf{c})$)

$\mathbf{a} \sim R_q \coloneqq \text{GenA}(\text{seed}_a)$
$\mathbf{w} \coloneqq \mathbf{az} - \mathbf{tc}$
if $\mathbf{c} = \text{H}(\text{Round}(\mathbf{w}), \mu)$ **then**
    return 1
else
    return 0
end if