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

Yuejun Liu[1,2], Yongbin Zhou[1,2]**, Shuo Sun[1,2], Tianyu Wang[1,2], and Rui Zhang[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}@iie.ac.cn

Abstract. Leakage during the signing process, including partial key exposure and partial (or complete) randomness leakage, may be devastating for the security of digital signatures. In this work, we consider the security of lattice-based Fiat-Shamir signatures in the presence of randomness leakage. Based on a connection with the ILWE problem introduced by Bootle et al. at Asiacrypt 2018, we show that the key recovery attack with partial randomness leakage can be reduced to a variant of ILWE (We call it FS-ILWE in this work). The ILWE problem is the problem of recovering the secret vector $\mathbf{s}$ given polynomially many samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ and is proven solvable if the error e is not superpolynomially larger than the inner product $\langle \mathbf{a}, \mathbf{s} \rangle$, whereas in the FS-ILWE $\mathbf{a}$ is a sparse vector with a fixed number of non-zero elements, which is either 1 or -1. With one nice probability property that the expectation and covariance of any two coefficients of $\mathbf{a}$ are zeros, we show that FS-ILWE can also be solved in polynomial time.
Consequently, many lattice-based Fiat-Shamir signatures can be totally broken with only one bit leakage of randomness per signature. Our attack has been validated by conducting a series of experiments on two efficient NIST PQC submissions, Dilithium and qTESLA. The results indicate that the secret key of Dilithium and qTESLA can be recovered within seconds by running our method on an ordinary PC desktop.

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 parameters are kept hidden. However, when a cryptographic algorithm

* This work is supported in part by National Natural Science Foundation of China (No.61632020, No.61772520) and Beijing Natural Science Foundation (No. 4192067).

** Corresponding author.

is practically used, these parameters may be leaked to adversaries due to implementation, communication or other reasons. Taking digital signatures as example, leakage during the signing process, including partial key exposure and partial (or complete) randomness leakage, may be devastating for their security. For example, Heninger and Shacham [25] showed that the RSA secret key with small public parameters can be efficiently recovered given 27% random bits. And DSA whose key is 160-bit can be totally broken if only 3 least significant bits (LSBs) of randomness are known [33]. In this work we focus on the security of signatures in the presence of partial randomness leakage.

Howgrave-Graham and Smart [26] proposed the first partial randomness (i.e. nonce) leakage attack on DSA by reducing it to the closet vector problem (CVP), which can be solved using the Babai's nearest plane algorithm [6] together with the LLL lattice reduction algorithm [28]. However, their attack relied on several heuristic assumptions. Later, Nguyen and Shparlinski [33] presented the first provable attack on DSA with partial randomness leakage. More precisely, with about $\log^{1/2} q$ LSBs or most significant bits (MSBs), the secret key of DSA can be recovered in polynomial time. The main idea of their attack is mapping the partial leakage attack on DSA to a Hidden Number Problem (HNP) introduced in [13], which can be reduced to the shortest vector problem (SVP) and then solved with the lattice reduction algorithms. Nguyen and Shparlinski showed that their attack can apply to DSA-like signatures, including ECDSA [34] and Schnorr's signature [43].

Because of the similarity between DSA and Fiat-Shamir signatures, we wonder whether the randomness leakage attack in [33] is applicable to the Fiat-Shamir signatures [22] besides Schnorr's signature whose signatures are in the form of $z = y + sc \mod q$. Recall that in HNP it is to recover the hidden number α 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 a 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 the most promising candidate and has gained a lot of attention. There are five lattice-based signatures submitted to NIST, two of which follow the Fiat-Shamir paradigm: Dilithium [32] and qTESLA [10]. Specifically, Dilithium avoids the NTRU lattice and discrete Gaussian sampling for the security and the convenience of implementation, meanwhile, it still has high efficiency and small sizes of public key and signature. And qTESLA is provably secure in the Quantum Random Oracle Model (QROM). In contrast to the theoretical security, the security of lattice-based Fiat-Shamir signatures in the presence of randomness leakage is still open.

Now there is a natural question that whether partial randomness leakage attacks on Fiat-Shamir signatures based on other mathematical structures, such as the attack in [33], can apply to lattice-based Fiat-Shamir signatures. The answer is negative. 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 a HNP over lattice.

1.1 Our Contributions

In this work, we present a new polynomial time key recovery attack on lattice-based Fiat-Shamir signatures by leaking only one bit of randomness used in the signing process. More precisely, the Fiat-Shamir signature is computed as $\mathbf{z} = \mathbf{y} + \mathbf{sc}$. 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 $\mathbf{c}$. We show that if the $(l+1)$-th bit of y is leaked, one can recover the secret key of Fiat-Shamir signatures over lattice, where l is the leakage bound satisfying $\Pr[|\langle \mathbf{s}, \bar{\mathbf{c}} \rangle| < 2^{l-1}] > 99\%$. Our attack still works if leakage occurs at any position between the $(l+1)$-th bit and the MSB of any coefficient of randomness. However, in the case of leaking higher-order bit, one needs more signatures to recover the secret key. Roughly speaking, almost four times as many signatures theoretically are necessary if the leakage position is shifted to left by one bit.

The main idea of our attack is reducing the key recovery attack with leakage to the Fiat-Shamir integer learning with error (FS-ILWE) problem, which is a variant of the ILWE problem [14]. The ILWE problem is the problem of recovering the secret vector $\mathbf{s}$ given polynomially many samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ and is proven solvable if the error e is not superpolynomially larger than the inner product $\langle \mathbf{a}, \mathbf{s} \rangle$, whereas in the FS-ILWE $\mathbf{a}$ is a sparse vector with a fixed number of non-zero elements, which is either 1 or -1. With the nice probability property that the expectation and covariance of any two coefficients of $\mathbf{a}$ are zero, we show that FS-ILWE can also be solved in polynomial time.

We choose Dilithium and qTESLA as two cases of study to verify our attack, where the former is based on module lattice and the latter is based on ring. Despite the underlying structure, their signatures are in the form of $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, hence we can perform randomness leakage attacks on Dilithium and qTESLA directly. Note that in Dilithium, because of the module lattice, the secret key is a matrix and each column corresponds to a random n-dimensional polynomial vector. Therefore, the attack on Dilithium can be reduced to an FS-ILWE problem whose dimension is as small as the degree of the underlying ring. From this perspective, module lattice is more vulnerable to our attack than ring at the same security level.

Also note that the secret key consists of two components: $\mathbf{s}_1$ and $\mathbf{s}_2$. However, we can only recover $\mathbf{s}_1$ via our attack since the signature $\mathbf{z}$ is the proof knowledge of $\mathbf{s}_1$ and the proof of knowledge of $\mathbf{s}_2$ is removed to reduce the signature size in Dilithium and qTESLA. Generally, for most of the lattice-based Fiat-Shamir signatures, such as qTESLA, $\mathbf{s}_2$ can be recovered with the public key $\mathbf{t} = \mathbf{A}\mathbf{s}_1 +$

$\mathbf{s}_2$ by solving a system of linear equations. However, since the public key is compressed, this method fails in Dilithium. Fortunately, recent works [15,41] proposed a signing algorithm to generate signatures only with the knowledge of $\mathbf{s}_1$. In other words, recovering $\mathbf{s}_1$ is sufficient for existential forgery attacks on Dilithium.

Our attack has been validated by conducting experiments on Dilithium and qTESLA. There are three types of experiments. As discussed above, the leakage bound l affects the success rate and the number of required signatures in our attack. Hence, we first determine the suitable l for each parameter set of Dilithium and qTESLA statistically. The results show that l for qTESLA is generally larger than that for Dilithium. Then we perform leakage attacks on both signatures by leaking the $(l+1)$-th bit of any coefficient of randomness. It takes only several seconds to recover the secret key of Dilithium on an ordinary desktop, while it takes about hundreds of seconds for qTESLA. The results also support the conclusion that Dilithium is easier to attack than qTESLA in our case due to the module lattice structure. Another interesting conclusion is that the difficulty of our attack is opposite to the difficulty of lattice reduction when the dimension n is fixed. For example, in Dilithium, the higher the security level claimed, the easier our attack is. Finally, we conduct experiments on Dilithium to recover the secret key with high-order leakage bit. The results indicate that if the leakage position is shifted to left by one bit, roughly two to five times signatures are required to recover the secret key.

1.2 Overview of Our Attack

Our attack stems from an observation that the Fiat-Shamir signatures over lattice look like ILWE samples. Specially, in the lattice-based Fiat-Shamir signature, the signature is computed as $\mathbf{z} = \mathbf{y} + \mathbf{c}\mathbf{s}$. 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 a sample of the ILWE problem. As shown in [14], the ILWE problem can be solved with high probability by the least squares method then 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, $\bar{\mathbf{c}}$ is a sparse vector whose non-coefficient is either 1 or -1, making σ_a very small. Worse still, in the lattice-based Fiat-Shamir signatures, we generally choose a large y to mask the secret key $\mathbf{s}$, which further increases the difficulty of solving the ILWE problem. Finally, 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.

We overcome the above two technique hurdles by throwing away the MSBs of the randomness and partially leaking the randomness respectively, corresponding to the first two steps of our attack. More specially, since y is much larger than $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, only the low-order bits of z contain information of $\mathbf{s}$. Hence, in the first step, we throw away the MSBs of y, z and only concentrate on their LSBs to reduce the error. In the second step, we establish the connection between the signature z and the secret key $\mathbf{s}$ by leaking one bit of randomness, which is also used to remove the module introduced in the first step. Assuming $\|\mathbf{sc}\|_\infty < 2^l$, the FS-ILWE problem in our attack is given as follows: $[z]_{2^l} \pm d \cdot 2^l = [y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, where $\mathbf{a} = \bar{\mathbf{c}}, e = [y]_{2^l}$ and $b = [z]_{2^l} \pm d \cdot 2^l$.

$[y]_{2^l}$ and $[z]_{2^l}$ correspond to the l LSBs of y and z. Note that we need to leak the $(l+1)$-th bit of y to judge whether the sum of $[y]_{2^l}$ and $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ 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 $d = 1$. The remaining thing is determining the sign of d which depends on the value of $[z]_{2^l}$ in case of $d = 1$, i.e. determining the overflow is caused by a carry or a borrow. If we choose l satisfying $\Pr[|\langle \mathbf{s}, \bar{\mathbf{c}} \rangle| < 2^{l-1}] > 99\%$, the probability of guessing b correctly is more that 99%. If we guess wrong, extra error will be introduced. Hence, with the probability more than 99%, the error term is exactly $[y]_{2^l}$. Thus, we obtain an FS-ILWE sample and key recovery attack with leakage is reduced to an FS-ILWE problem where $\mathbf{a}$ has special structure. We also show that FS-ILWE can be solved using least squares regression in polynomial time in Section 3.3. The case of leaking the high-order bit of randomness is similar and the only difference is the error distribution which is $[y]_{2^{t-1}}$ instead and t is the leakage position between the $(l+1)$-th bit and the MSB of randomness.

1.3 Related Work

Leakage Attacks on (EC)DSA. The works [26,33,34] have shown that attacks on DSA-like signature schemes with partial known randomness can be mapped to a HNP problem, which can be reduced to CVP or SVP and solved by lattice reduction techniques [6,28,44,17]. Based on the idea, a series of works estimated the security of implementations of DSA and ECDSA in OpenSSL [16,1,2,9,45,3]. Almost all of them used the cache-based side-channel attacks to extract the leaked information except [2], which used a remote timing attack to obtain the MSBs of the ECDSA randomness. With the FLUSH+RELOAD attack [24,46], one can recover the secret key of DSA [1,36,4] and ECDSA [16,9,45,3] by partially leaking the randomness. And [29] reduced the number of required LSB of randomness for 160-bit DSA key from 3 to 2 by proposing a new lattice reduction technique.

Besides leakage, other information of randomness can also be utilized to attack. [8] showed that one can recover the secret key of DSA if the randomness is generated by Knuth's linear congruential pseudorandom number generator. And subsequent works also proposed lattice attacks on DSA and ECDSA with the relation between randomness, such as randomness have a congruence relation [11,39,40] or share some bits [21].

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 start to investigate the implementation security against leakage and fault attacks. [23] proposed the first side-channel attack on the Bimodal Lattice Signature Scheme (BLISS)

[19], which follows the Fiat-Shamir paradigm. The target of their attack is the Gaussian sampling algorithm used to generate the randomness polynomial and the main idea is leaking almost the entire $\mathbf{y}$ using the FLUSH+RELOAD cache-attack. With the signature $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, the secret key $\mathbf{s}$ can be recovered via basic linear algebra or lattice reduction technique. Later, [38] extended the cache attack to BLISS-B [18]. Another work of randomness leakage attack is [20], whose attack target is also the Gaussian sampling algorithm. They leaked the entire value of randomness $\mathbf{y}$ and hence the secret key using the branch tracing technique.

The main reason of randomness leakage is that the randomness is generated by a variable-time Gaussian sampling algorithm, which is vulnerable to side-channel attacks. Aware of this, the later signatures such as Dilithium and qTESLA employ the uniform distribution instead of Gaussian distribution to avoid above side-channel attacks. Hence, Dilithium and qTESLA are secure against these randomness leakage attacks because large randomness leakages are no longer feasible. Our attack requires only one bit leakage per signature, and one bit leakage is easier to obtain in practice. That is to say, our work applies to lattice-based Fiat-Shamir signatures whose randomness follows both Gaussian distribution and uniform distribution.

Besides the Gaussian sampling algorithm, [20] also considered two other leakage sources: the rejection sampling algorithm and the polynomial multiplication $\mathbf{s}_1\mathbf{c}$. 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. The latter can be used to recover the secret key $\mathbf{s}_1$ directly by using a traditional differential power analysis (DPA) or EMA. 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. [14] 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.

It is worth noting that although our attack is mapped to the FS-ILWE problem which is a variant of ILWE problem in the attack of [14], their attack needs to obtain the noisy linear leakage by side-channel attacks on the rejection sampling, while our attack is an abstract 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 cannot apply to Dilithium and qTESLA because of the uniform distribution.

[41] adapted the side-channel attack on polynomial multiplication to Dilithium, however, one can only recover $\mathbf{s}_1$. Since the public key of Dilithium is compressed, the secret key $\mathbf{s}_2$ is still unknown. They proposed an alternate signing algorithm with only $\mathbf{s}_1$.

In addition to leakage attacks, countermeasure such as shuffling on Gaussian sampling [42] was proposed. However, [37] proposed an attack on the shuffling countermeasure.

Discussion about the Attack Model. To verify the reality of our attack model, in Appendix F we provide experiments which describe how to get a single specific bit of randomness in practice. Indeed, recent lattice-based Fiat-Shamir signatures such as Dilithium and qTESLA employ the uniform distribution instead of Gaussian distribution to generate randomness to avoid side-channel attacks on Gaussian sampling, making it hard to get information during the randomness generation process. However, our attack target can be any step that involve the manipulation of randomness $\mathbf{y}$, for example, the operation $\mathbf{z} = \mathbf{y} + \mathbf{sc}$ in Step 7 of Algorithm 6. The experiments show that we can recover the required leakage bit of randomness with probability 1. Since no complex methods and special techniques are required in our experiments, we believe the assumption that an adversary can recover a specific bit of y is realistic.

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 $\mathbf{x} = (x_0, x_1, \ldots, x_n)^T \in \mathbb{R}^n$ 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 $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ 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 $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=0}^{d} a_i b_i = \mathbf{a}^T \mathbf{b}$. 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 $\|\mathbf{A}\|_p^{\mathrm{op}}$ of $\mathbf{A}$ with respect to the p-norm is given by

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

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 at $[a]_{2^l} = a \pmod{2^l}$ when $a \geq 0$ and $[a]_{2^l} = (|a| \pmod{2^l})$ when $a < 0$. We extend the definition to vectors: for a vector $\mathbf{v} = (v_1, \ldots, v_n)$, $[\mathbf{v}]_{2^l}$ 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 [14] 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 $\mathbf{x}$[3].

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* $\tau' = \|\mathbf{A}^T\|_2^{\mathrm{op}} \cdot \tau$.

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 [14] 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* $\mathbf{s} \in \mathbb{Z}^n$ *recover the vector* $\mathbf{s}$.

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. [14] 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 Fiat-Shamir signatures by leaking the randomness used in the signing process. In a Fiat-Shamir signature whose form is $\mathbf{z} = \mathbf{y} + \mathbf{sc}$, the random oracle output $\mathbf{c}$ and the signature $\mathbf{z}$ are known and the secret key $\mathbf{s}$ and the randomness $\mathbf{y}$ are unknown. For each coefficient z of $\mathbf{z}$, it is obtained by $z = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle$, where $\bar{\mathbf{c}}$ is

[3] For completeness, we provide proofs of Lemma 4 and Lemma 5 in Appendix A, which are almost the same as that in [14].

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 $\mathbf{s}$ by the least squares method. Firstly, the distribution of signature is not related to the distribution of $\mathbf{s}$, so we show how to recover $\mathbf{s}$ with only one bit leakage in Section 3.1. Another obstacle is our attack can be reduced to an FS-ILWE problem, which is different from ILWE defined in [14], but we show FS-ILWE problem is also solvable with linear regression in Section 3.3.

3.1 Description of Our Attack

The crux of our attack relies on two observations: the infinity norm of $\mathbf{sc}$ is smaller than that of $\mathbf{y}$ so that $\mathbf{sc}$ only effects the low-order bits of the signature $\mathbf{z}$ and the Fiat-Shamir signatures over lattice seem like ILWE samples. The former shows that the high-order bits of $\mathbf{y}$ are not necessary for our attack (corresponding to Section 3.1) and the latter allows us to throw away the low-order bits to further reduce the number of required leakage bits (corresponding to Section 3.1).

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 is 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. Taking Dilithium as an example, with the recommended parameters whose quantum security level is 125-bit, $\|\mathbf{sc}\|_\infty$ is less than 6 bits [4]. 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 $\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$.

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 signature over lattice 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 [14], such a problem can be solved in polynomial time using statistical learning technique. Combining with the first observation

[4] In Dilithium, the original β is 8 bits so that $\|\mathbf{sc}\|_\infty \leq \beta$ except 2^{-80} probability. 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.

above, we can reduce the leakage attack on the lattice-based Fiat-Shamir signatures to an ILWE-like problem with relatively small errors and solve it using the least squares method. Assuming $\|\mathbf{sc}\|_\infty < 2^l$, the signature can be rewritten as:

$$
\begin{aligned}
& z && = y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle && (2)\\
\Rightarrow\quad & z \bmod 2^l && = (y + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \pmod{2^l} = (y \bmod 2^l + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \pmod{2^l} && (3)\\
\Rightarrow\quad & [z]_{2^l} \pm d \cdot 2^l && = [y]_{2^l} + \langle \mathbf{s}, \bar{\mathbf{c}} \rangle && (4)
\end{aligned}
$$

where we let

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

(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. The problem is called the FS-ILWE problem in the rest of the paper. We will estimate the distribution of error term, denoted by $\chi_e^{(\mathbf{a},\mathbf{s})}$, in Section 3.1.

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 there is an overflow, there are three cases[5]:

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

Because both of 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

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

satisfying $\Pr[|\langle \mathbf{s}, \bar{\mathbf{c}} \rangle| < 2^{l-1}] > 99\%$ in actual experiments[6]. 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 $\chi_{\mathbf{e}}^{(\mathbf{a},\mathbf{s})}$ 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 obeys a public and fixed distribution, denoted by χ_z, including the discrete Gaussian distribution and the uniform distribution. To do this simply, we assume that z is a uniform distribution on $(-2^\gamma, 2^\gamma) \cap \mathbb{Z}$, then y is a uniform distribution on $(-2^\gamma - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle, 2^\gamma - \langle \mathbf{s}, \bar{\mathbf{c}} \rangle) \cap \mathbb{Z}$, denoted by $\chi_e^{(\mathbf{a},\mathbf{s})}$. Let the probability density function of y is $p_y(x)$, then the probability density function of $e = [y]_{2^l}$ is

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

It is easy to work out $\mathbb{E}([y]_{2^l}) = -\frac{2^l-1}{2^{\gamma+1}-1}\langle \mathbf{s}, \bar{\mathbf{c}} \rangle$ is close to 0, so we can approximately regard $[y]_{2^l}$ as subgaussian over a bounded interval $(-2^l, 2^l)$.

Taken together, the leakage attack in the presence of 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.

[6] 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. $\Pr[|\langle \mathbf{s}, \bar{\mathbf{c}} \rangle| < 2^{l-1}] > 99\%$.

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 from the t-th bit of $\mathbf{y}$ where $l+1 \leq t \leq k$ 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

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

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. Theoretically, whenever the leakage location is shifted to 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 coefficients that are either -1 or 1 with equal probability and the rest are 0. 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* $\{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq m}$ *from the FS-ILWE distribution* $\mathcal{D}_{\mathbf{s},B_h,\chi_e^{(\mathbf{a},\mathbf{s})}}$ *for some* $\mathbf{s} \in \mathbb{Z}^n$ *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 $\chi_e^{(\mathbf{a},\mathbf{s})}$. Hence, we need to approximate $\chi_e^{(\mathbf{a},\mathbf{s})}$ as subgaussian (See Section 3.1 for details), introducing a heuristic assumption. 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 $\mathbf{s} = \lceil\tilde{\mathbf{s}}\rfloor = (\lceil\tilde{s}_1\rfloor, \ldots, \lceil\tilde{s}_1\rfloor)$ and the FS-ILWE problem is solved [7]. 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 [27, 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* $i = 1, .., m$*.*

[7] 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.

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:
$$\begin{aligned}\mathbb{E}[c_{ij}c_{ik}] = &(\Pr[c_{ij} = 1, c_{ik} = 1] + \Pr[c_{ij} = -1, c_{ik} = -1]) - (\Pr[c_{ij} = 1, c_{ik} = -1] \\ &+ \Pr[c_{ij} = -1, c_{ik} = 1]) \\ &= (\frac{h \cdot (h-1)}{2n \cdot 2(n-1)} + \frac{h \cdot (h-1)}{2n \cdot 2(n-1)}) - (\frac{h \cdot (h-1)}{2n \cdot 2(n-1)} + \frac{h \cdot (h-1)}{2n \cdot 2(n-1)}) = 0\end{aligned}$$
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 [27, 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 $\mathbf{x}_i = \sqrt{n/h} \cdot \mathbf{c}_i (1 \leq i \leq m)$. 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):

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

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 $m \geq 144n(n \log(9) + \log(2^{1+\eta}))/(h\beta^2)$, there are $\sqrt{8} + \sqrt{\rho} \leq 3$, then we have:

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

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* $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ *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* $\eta \geq 1$*, if*

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

then the least squares estimator $\tilde{\mathbf{s}} = (\mathbf{A}^T\mathbf{A})^{-1} \cdot \mathbf{A}\mathbf{b}$ *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 $m \geq 4n(C_1 n + C_2\eta)/h$, 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:

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

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

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

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.

4 Two Cases of Study: Dilithium and qTESLA

Among five lattice-based signature schemes submitted to NIST, Dilithium and qTESLA follow the Fiat-Shamir paradigm and both are promising. Due to their structure, they are vulnerable to such partial randomness leakage attacks and the reason is explained blow.

4.1 Attacks on Dilithium

The Dilithium scheme is built via the "Fiat-Shamir with abort" structure [30,31] and includes several optimizations on top of the Bai-Galbraith scheme [7]. The security of Dilithium is based on the hardness of Module-LWE and Module-SIS problem, a flexible generalization of Ring-LWE and Ring-SIS. The signing algorithm is given by Algorithm 3 and we defer the whole description of Dilithium to Appendix D.

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

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

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

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 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 works [15,41] show that just

Algorithm 1 Sign($sk = (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}), \mu \in \mathcal{M}$)

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

knowing $\mathbf{s}_1$ is sufficient for existential forgery attack, so it is not necessary to recover $\mathbf{s}_2$ to forge valid signatures.

Obviously, recovering $\mathbf{s}_1$ in the presence of leaking $(l+1)$-th bit of any coefficient of $\mathbf{y}$ is exactly an FS-ILWE problem. Consider, for instance, the recommended parameters in which Dilithium achieves 125 bits security against quantum adversaries and 138 bits security against classical adversaries. In the FS-ILWE problem we reduced to, $\mathbf{a}$ 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 Dilithium is a signature based on MLWE, the secret key can be represented by a matrix (for example, a 256×4 matrix in the recommended parameter set) 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, which can be computed in parallel, so the time needed in this attack is the same as which in one 256-dimension FS-ILWE problem, however, we need 4 bits 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

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 as well. The main difference between Dilithium and qTESLA is the mathematical structure: Dilithium is based on the hardness of Module-LWE and

Module-SIS problem, while qTESLA is based on the hardness of Ring-LWE in $\mathbb{Z}_q[x]/(x^n+1)$. We defer the description of qTESLA to Appendix E.

Our partial randomness leakage attack can be adapted to qTESLA directly. Compared with Dilithium, attacks on qTESLA are even simpler 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, the public key of qTESLA is not compressed, hence another component of the signing key $\mathbf{e}$ can be recovered easily after $\mathbf{s}$ is known.

Besides, in a ring 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 in the case of one bit leakage is n times that required in the case of n bits leakages. Hence, although our attack in Section 3.1 describes how to recover the secret key with only one bit leakage per signature, for efficiency, we instead leak more than one bit in actual attacks on Dilithium and qTESLA. And 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 partial randomness leakage attack on Dilithium and qTESLA. Specially, we first describe key recovery attacks on Dilithium and qTESLA by leaking the $(l+1)$-bit of any coefficient of randomness in Section 5.2, then taking Dilithium as an example, we show how to perform such attack with leakage from other positions in Section 5.3.

	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	$\|\mathbf{y}_i\|_\infty \le \gamma_1 - 1$	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	$\|\mathbf{s}_i\|_\infty \le \eta$	7	6	5	3
β	$\|\mathbf{s}_{1,2}\mathbf{c}\|_\infty \le \beta$	330	285	235	145
classical security		58	100	138	174
quantum security		53	91	125	158

Table 1. Parameters for Dilithium

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 given in the parameter sets of Dilithium and qTESLA actually. However, in order to guarantee 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. And the smaller l is, the less signatures required to recover the secret key. Hence, we study the probability of $\|\mathbf{sc}\|_\infty < 2^l$ on different l statistically for Dilithium and qTESLA to find a suitable l.

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.

	$l = 5$	$l = 6$	$l = 7$	$l = 8$	$l = 9$	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 2. The probability of $\|\mathbf{s}_1\mathbf{c}\|_\infty \leq 2^l$ in Dilithium

		I	II	III	IV	V
	description	qTESLA-I	qTESLA-	qTESLA-	qTESLA-	qTESLA-
			III-speed	III-size	p-I	p-III
n	dimension	512	1,024	1,024	1,024	2,048
q	modulus	4,205,569	8,404,993	4,206,593	485,978,113	1,129,725,953
h	weight of $\mathbf{c}$	30	48	48	25	40
B	$\|\mathbf{y}_i\|_\infty \leq B$	$2^{20} - 1$	$2^{21} - 1$	$2^{20} - 1$	$2^{21} - 1$	$2^{23} - 1$
σ	sk std. dev.	23.78	10.2	8.49	8.5	8.5
L_S	$\|\mathbf{sc}\|_\infty \leq L_S$	1,586	1,233	910	554	901
classical security		104	178	188	132	247
quantum security		97	164	169	123	270

Table 3. Parameters for qTESLA

For qTESLA, the authors specify five parameter sets, which are displayed in Table 3. The first three parameter sets, namely qTESLA-I, qTESLA-III-speed and qTESLA-III-size, are chosen heuristically and the last two parameter sets, namely qTESLA-p-I and qTESLA-p-III, are chosen according to security reduction. Since the maximum infinity norm of $\mathbf{sc}$ in Table 3 is 1586, which is less than 2^{11}, we compute the probability of $\|\mathbf{sc}\|_\infty < 2^l$ when $l = 6, 7, 8, 9, 10, 11$ for each set by choosing $10,000$ signatures (corresponding to $10,240,000$ coefficients of $\mathbf{sc}$) randomly. From the statistical results in Table 4, the leakage bound l is set to $10, 9, 9, 8, 9$ respectively and is larger than l in Dilithium. Worse still, the dimension of n is larger and the number of non-zero coefficients in $\mathbf{c}$ is less, leading that the attack on qTESLA is much harder than that on Dilithium. Hence, in next section we only consider the qTESLA-II (corresponding to qTESLA-III-speed) and qTESLA-III (corresponding to qTESLA-III-size) parameter sets for the sake of time and memory, but our attack can apply to other parameter sets.

	$l = 6$	$l = 7$	$l = 8$	$l = 9$	$l = 10$	$l = 11$	leakage bound l
qTESLA-I	0.39013	0.69467	0.96014	0.99996	1	1	10
qTESLA-II	0.62822	0.92709	0.9997	1	1	1	9
qTESLA-IIII	0.77561	0.98525	0.99999	1	1	1	9
qTESLA-IV	0.86942	0.99762	1	1	1	1	8
qTESLA-V	0.77011	0.984125	0.99999	1	1	1	9

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

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 used in Dilithium and qTESLA. Our attack consists of three steps: generating signatures with the $(l+1)$-th bit leakages, then reducing to an FS-ILWE problem and finally solving the 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 a 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. Take Dilithium-III as an example, we show the number of FS-ILWE samples required in case of leaking one bit of randomness per signature is almost equal to that in case of leaking one bit of randomness per coefficient. 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 can be seen from Table 5, the difference of interquartile mean is about 6.32%, meaning that the number of FS-ILWE samples in two cases is not much 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.

Number of leakage bits per signature	Min	LQ	IQM	UQ	Max
256	849,664	1,011,584	1,342,080	1,453,184	1,716,224
1	1,055,232	1,152,640	1,432,576	1,524,608	1,583,616
DIF	19.48%	12.24%	6.32%	4.68%	-8.37%

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

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, thus the complexity of computing $\mathbf{C}^T\mathbf{C}$ and $\mathbf{C}^T\mathbf{b}$ is reduced from $O(mn^2)$ and $O(mn)$ to $O(mh^2)$ and $O(mh)$. Moreover, instead of computing $\mathbf{C}^T\mathbf{C}$ and $\mathbf{C}^T\mathbf{b}$ directly, we use the block matrix strategy and compute block by block to avoid memory overflow.

Now we turn to concrete experiments on Dilithium and qTESLA. For Dilithium, we consider all parameter sets and for each set, we perform 12 trails. 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, it is not necessary to use so many signatures. We can recover most of coefficients with fewer signatures and then recover the entire secret key by brute force or other techniques. Due to space limitations, there is no further discussion here.

	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	2,561	3,053	3,284	3,778	5,511

Table 6. Numbers of signatures required to recover the secret key of Dilithium (minimum, lower quartile, interquartile mean, upper quartile, maximum)

Interesting enough, we conclude that the difficulty of our attack is opposite to the difficulty of lattice reduction from Table 6. The higher the security level claimed, the less FS-ILWE samples required and the easier our attack is. The 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-zeros of $\mathbf{c}$ is fixed, l is positively related to the value of the secret key $\mathbf{s}$. For Dilithium-I through Dilithium-IV, the secret key is getting smaller and smaller. Thus, in our attack, Dilithium-I is the most difficult and Dilithium-IV is the easiest.

In table 7, we present the running time for our attack on Dilithium. Since generating signatures in the first step is not our business, we omit the time of generating signatures here. Moreover, the running time of solving a linear system consisting of n equations is constant if the dimension n is fixed and negligible [8] 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 $\mathbf{C}^T\mathbf{C}$ and $\mathbf{C}^T\mathbf{b}$ in the third step due to the large dimension m and cover 99.97% of the running time.

	Time for FS-ILWE samples (ms)	Time for $\mathbf{C}^T\mathbf{C}$ and $\mathbf{C}^T\mathbf{b}$ (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

Table 7. Average running time to recover the secret key of Dilithium

Similarly, we perform our attack on qTESLA. For each parameter set, we perform 12 trails. Results about the minimum numbers of required samples and corresponding running time are displayed in Table 8 and 9. The total time of recovering the secret key of qTESLA is within hundreds of seconds. Different from Dilithium, we only consider two parameter sets qTESLA-II and qTESLA-III. We notice that even though the claimed security levels of qTESLA-II and qTESLA-III are higher than that of Dilithium, attacks on qTESLA are more difficult, mainly because of the larger dimension n, the more sparse 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 more sparse $\mathbf{c}$ when n is larger. Therefore, we may conclude that at the same security level, module lattice is

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

more vulnerable to our attack than ring because the dimension of the underlying ring of module lattice is generally smaller. The experimental results also verify that.

	Min	LQ	IQM	UQ	Max
qTESLA-II	175,119	199,354	261,871.5	277,540.5	315,051
qTESLA-III	150,254	161,704.5	174,078.5	234,399	327,166

Table 8. Numbers of signatures required to recover the secret key of qTESLA (minimum, lower quartile, interquartile mean, upper quartile, maximum)

	Time for FS-ILWE samples (ms)	Time for $\mathbf{C}^T\mathbf{C}$ and $\mathbf{C}^T\mathbf{b}$ (s)	The total time (s)
qTESLA-II	229.66	846.16	846.39
qTESLA-III	145.56	613.48	613.63

Table 9. Average running time to recover the secret key of qTESLA

In table 10, we provide numbers of required leakage bits and signatures required to attack DSA, ECDSA, Dilithium and qTESLA. And it can be seen that lattice-based Fiat-Shamir signatures are currently easier to attack because less signatures are required when leaking one bit of the randomness at the almost 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.

	Classical security	Leakage bits	Signatures	Work
DSA	160	2	100	[29]
DSA	160	$\log 3 \approx 1.58$	2^{22}	[12]
ECDSA	160	1	2^{33}	[5]
Dilithium	174	1	2^{20}	our work
qTESLA	178	1	2^{28}	our work

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

5.3 Attacking Dilithium by Leaking High-Order Bits

Assume the length of coefficients of randomness $\mathbf{y}$ is k. Section 3.2 shows that we can recover the secret key of lattice-based Fiat-Shamir signatures with one leakage bit of any position between $l+1$ and k of any coefficient of $\mathbf{y}$. In this section, we perform attacks on Dilithium to show how to recover the secret key with high-order leakage bits. 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 at different but known positions. According to Table 2, for four parameter sets of 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 displayed in Figure 1.

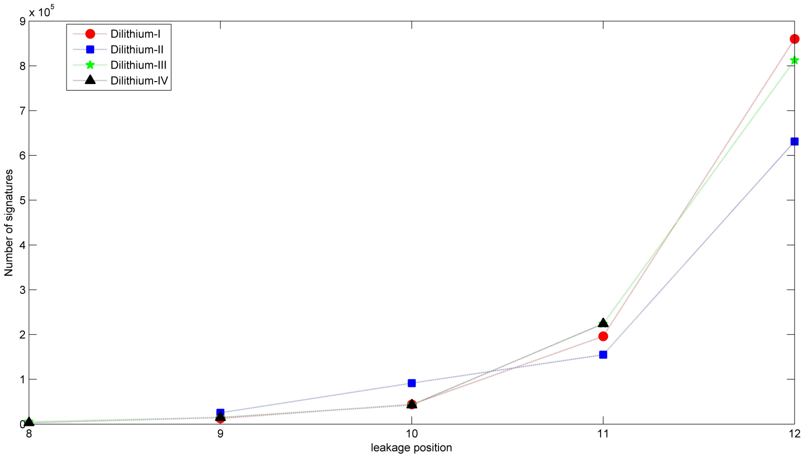

Fig. 1. Number of signatures in the presence of high-order bit leakage

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. For example, when $l = 9$, the number of samples required to recover the secret keys of Dilithium-I, Dilithium-II, Dilithium-III and Dilithium-IV are 44018, 91493, 42791 and 42979 respectively. The results are consistent with the theoretical results. Fix 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 $1,000,000$ we set in advance. Another conclusion that can be drawn from Figure 1 is that when the leakage position is shifted left by one, the number of signatures required is 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 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 [35]. 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.

6 Conclusion

In this work we present a polynomial time attack on lattice-based Fiat-Shamir signatures with only one bit leakage per signature. We show 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. We choose Dilithium and qTESLA as two cases of study to verify our attack.

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

Our attack is applicable to most of lattice-based Fiat-Shamir signatures except BLISS. To improve the success rate of the rejection sampling, BLISS use a bimodal Gaussian distribution and the signature is $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{sc}$ where b 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.

References

1. Onur Acıiçmez, Billy Bob Brumley, and Philipp Grabher. New results on instruction cache attacks. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, pages 110–124, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
2. Onur Acıiçmez, Billy Bob Brumley, and Philipp Grabher. Remote timing attacks are still practical. In Vijay Atluri and Claudia Diaz, editors, *ESORICS*, pages 355–371, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
3. Thomas Allan, Billy Bob Brumley, Katrina Falkner, Joop van de Pol, and Yuval Yarom. Amplifying side channels through performance degradation. In Stephen Schwab, William K. Robertson, and Davide Balzarotti, editors, *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 422–435. ACM, 2016.
4. Jiji Angel, R. Rahul, C. Ashokkumar, and Bernard Menezes. DSA signing key recovery with noisy side channels and variable error rates. In Arpita Patra and Nigel P. Smart, editors, *INDOCRYPT*, pages 147–165, Cham, 2017. Springer International Publishing.
5. Diego F. Aranha, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, Mehdi Tibouchi, and Jean-Christophe Zapalowicz. GLV/GLS decomposition, power analysis, and attacks on ECDSA signatures with single-bit nonce bias. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT*, pages 262–281, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
6. Lszl Babai. On lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

7. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *CT-RSA*, pages 28–47, Cham, 2014. Springer International Publishing.
8. Mihir Bellare, Shafi Goldwasser, and Daniele Micciancio. "pseudo-random" number generation within cryptographic algorithms: The DDS case. In Burton S. Kaliski, editor, *CRYPTO*, pages 277–291, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
9. 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 Lejla Batina and Matthew Robshaw, editors, *CHES*, pages 75–92, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
10. 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/`.
11. Ian F. Blake and Theodoulos Garefalakis. On the security of the digital signature algorithm. *Designs, Codes and Cryptography*, 26(1-3):87–96, 2002.
12. Daniel Bleichenbacher. On the generation of DSS one-time keys. Manuscript. The result was presented at the Monteverita workshop in March 2001.
13. Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In Neal Koblitz, editor, *CRYPTO*, pages 129–142, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
14. Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT*, pages 494–524, Cham, 2018. Springer International Publishing.
15. Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(3):21–43, 2018.
16. Billy Bob Brumley and Risto M. Hakala. Cache-timing template attacks. In Mitsuru Matsui, editor, *ASIACRYPT*, pages 667–684, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
17. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
18. Léo Ducas. Accelerating Bliss: the geometry of ternary polynomials. Cryptology ePrint Archive, Report 2014/874, 2014. `https://eprint.iacr.org/2014/874`.
19. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *CRYPTO*, pages 40–56, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
20. 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, New York, NY, USA, 2017. ACM.
21. Jean-Charles Faugère, Christopher Goyet, and Guénaël Renault. Attacking (EC)DSA given only an implicit hint. In Lars R. Knudsen and Huapeng Wu, editors, *SAC*, pages 252–274, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

22. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.
23. 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 Benedikt Gierlichs and Axel Y. Poschmann, editors, *CHES*, pages 323–345, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
24. David Gullasch, Endre Bangerter, and Stephan Krenn. Cache games–bringing access-based cache attacks on AES to practice. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 490–505. IEEE, 2011.
25. Nadia Heninger and Hovav Shacham. Reconstructing RSA private keys from random key bits. In Shai Halevi, editor, *CRYPTO*, pages 1–17, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
26. Nick A Howgrave-Graham and Nigel P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23(3):283–290, 2001.
27. 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.
28. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
29. Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In Ed Dawson, editor, *CT-RSA*, pages 293–309, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
30. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT*, pages 598–616, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
31. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, pages 738–755, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
32. 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`.
33. Nguyen and Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15(3):151–176, 2002.
34. 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.
35. National Institute of Standards on Technology (NIST). Post-quantum cryptography standardization. `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization`.
36. Cesar Pereida García, Billy Bob Brumley, and Yuval Yarom. Make sure DSA signing exponentiations really are constant-time. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *CCS*, pages 1639–1650. ACM, 2016.
37. Peter Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *INDOCRYPT*, pages 153–170, Cham, 2016. Springer International Publishing.
38. 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, New York, NY, USA, 2017. ACM.

39. Dimitrios Poulakis. Some lattice attacks on DSA and ECDSA. *Applicable Algebra in Engineering, Communication and Computing*, 22(5-6):347–358, 2011.
40. Dimitrios Poulakis. New lattice attacks on DSA schemes. *Journal of Mathematical Cryptology*, 10(2):135–144, 2016.
41. 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`.
42. Markku-Juhani O. Saarinen. Arithmetic coding and blinding countermeasures for lattice signatures. *Journal of Cryptographic Engineering*, 8(1):71–84, 2018.
43. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO*, pages 239–252, New York, NY, 1990. Springer New York.
44. Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.
45. Joop van de Pol, Nigel P. Smart, and Yuval Yarom. Just a little bit more. In Kaisa Nyberg, editor, *CT-RSA*, pages 3–21, Cham, 2015. Springer International Publishing.
46. Yuval Yarom and Katrina Falkner. FLUSH+ RELOAD: A high resolution, low noise, L3 cache side-channel attack. In Kevin Fu and Jaeyeon Jung, editors, *USENIX Security Symposium*, pages 719–732, San Diego, CA, 2014. USENIX Association.

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, $|\mu| \leq \|\mathbf{A}^T\|_2^{\mathrm{op}}$, 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 $\mathbf{b} = \mathbf{Cs} + \mathbf{e}$ 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 $\mathbf{b} = \mathbf{Cs}' + \mathbf{e}'$ 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 [9], 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 $\mathbf{s}' = (1 - \frac{2^l-1}{2^{\gamma+1}-1})\mathbf{s}$.

[9] 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.

D Description of Dilithium

The Dilithium scheme is built via the “Fiat-Shamir with abort” structure and includes several optimizations on top of the Bai-Galbraith scheme [7]. The security of Dilithium is based on the hardness of Module-LWE and Module-SIS problem, a flexible generalization of Ring-LWE and Ring-SIS. The Dilithium scheme is given by Algorithms 2-4.

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 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 4.

Algorithm 2 KeyGen()

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

E Description of qTESLA

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 as well. The main difference between Dilithium and qTESLA is the mathematical structure: Dilithium is based on the hardness of Module-LWE and Module-SIS problem and qTESLA is based on the hardness of Ring-LWE in $\mathbb{Z}_q[x]/(x^n+1)$. The simplified qTESLA scheme is given by Algorithm 5-7[10].

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

Algorithm 3 Sign($sk = (\rho, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}), \mu \in \mathcal{M}$)

```
1:  $\mathbf{A} \sim R_q^{k \times l} := \mathrm{Sam}(\rho)$
2:  $\mathbf{t}_1 := \mathrm{Power2Round}_q(\mathbf{t}, d)$
3:  $\mathbf{t}_0 := \mathbf{t} - \mathbf{t}_1 \cdot 2^d$
4:  $\mathbf{r} \leftarrow \{0,1\}^{256}$
5:  $\mathbf{y} \sim S_{\gamma_1 - 1}^{l} := \mathrm{Sam}(\mathbf{r})$
6:  $\mathbf{w} := \mathbf{Ay}$
7:  $\mathbf{w}_1 := \mathrm{HighBits}_q(\mathbf{w}, 2\gamma_2)$
8:  $\mathbf{c} := \mathrm{H}(\rho, \mathbf{t}_1, \mathbf{w}_1, \mu)$
9:  $\mathbf{z} := \mathbf{y} + \mathbf{cs}_1$
10: $(\mathbf{r}_1, \mathbf{r}_0) := \mathrm{Decompose}_q(\mathbf{w} - \mathbf{cs}_2, 2\gamma_2)$
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: $\mathbf{h} := \mathrm{MakeHint}_q(-\mathbf{ct}_0, \mathbf{w} - \mathbf{cs}_2 + \mathbf{ct}_0, 2\gamma_2)$
15: if $\|\mathbf{ct_0}\|_\infty \geq \gamma_2$ or the number of 1's in $\mathbf{h}$ is greater than $\omega$ then
16:     goto 4
17: end if
18: return $\sigma = (\mathbf{z}, \mathbf{h}, \mathbf{c})$
```

Algorithm 4 Verify($vk = (\rho, \mathbf{t}_1), \mu \in \mathcal{M}, \sigma = (\mathbf{z}, \mathbf{h}, c)$)

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

Algorithm 5 KeyGen()

```
1: $\mathrm{seed}_a \leftarrow \{0,1\}^{256}$
2: $\mathbf{a} \sim R_q := \mathrm{GenA}(\mathrm{seed}_a)$
3: while $\mathbf{s}$ and $\mathbf{e}$ do not fulfill certain criteria do
4:     $\mathbf{s} \sim R_q \leftarrow D_\sigma, \mathbf{e} \sim R_q \leftarrow D_\sigma$
5: end while
6: $\mathbf{t} = \mathbf{as} + \mathbf{e} \bmod q$
7: return $(vk = (\mathrm{seed}_a, \mathbf{t}), sk = (\mathbf{s}, \mathbf{e}, \mathrm{seed}_a))$
```

Algorithm 6 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 7 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
```

F Practical Experiments

In this section, we show how to come up with the required bit of randomness in practice[11].

F.1 A template Attack on Polynomial Addition

We describe practical experiments about how to get 1 bit information of y through power leakages in this section. Our attack target is the signing algorithm. A software platform is used to implement the sensitive operation related to y (i.e. $\mathbf{z} = \mathbf{y} + \mathbf{sc}$), and the power leakages of the devices are measured by an oscilloscope (Agilent DSO9104A). The software platform is an 8051 microcontroller (MCU) clocked at 11.0592MHz. An addition implementation runs on it. 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 100,000 samples, which are around the sensitive operation. We have 2 groups of power traces, one is collected without filter and the other is collected with a lower pass filter (BLP-90+).

First, we use T-test to detect the leakages for 8-th bit of y with 4,000 traces gathered in each group. The leakages collected with a BLP get much more obvious, and the leakages (with a lower pass filter) are shown in Fig 2.

[11] Source codes are available at https://www.dropbox.com/sh/z4a3miy8lqvx46z/AACdtKbkCyTETWM4p202JWj9a?dl=0.

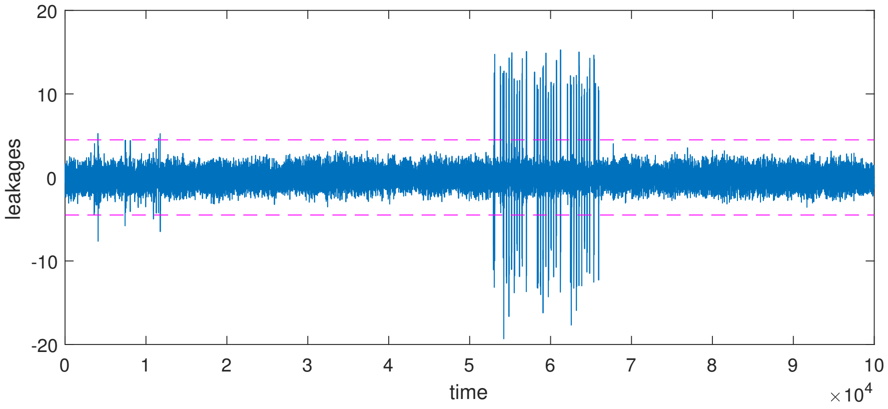

Fig. 2. The leakages of 8-th bit (with a lower pass filter).

In practical attacking, we use Template Attack (TA, one of profiling attacks) to recover the 8-th bit of y. Simply, in each group, we use lots of traces (no more than 4000) to profile for $bit = 0$ and $bit = 1$, then use other 239 traces to verify how many bits can be recovered correctly. The attack results are shown in Table 11.

Table 11. The results of TA aiming to recover 8-th bit of y (with 263 Points$^{\#}$ of interest used).

Setup	Traces$^{\#}$ for profiling	Success Rate
no filter	600	48.50%
	1800	53.5%
	4000	43.50%
BLP filter	600	74.5%
	1800	94.5%
	4000	**100%**

As shown in Table 11, while 4000 traces (with BLP filter) are used to profile, we can totally recover the 8-th bit in our setup. In fact, other bits of y can be recovered similarly, which are shown in Table 13.

F.2 A Template Attack on Randomness Generation

In this section, we describe how to get 1 bit information of y targeting the loading and moving process of y though power leakages. A software platform is used to implement the randomness generation operation. The implementation environment, acquisition equipment and parameters setting are the same as those in Section F.1.

Table 12. The results of TA on polynomial addition aiming to recover any bit of y.

i-th bit	Traces$^{\#}$ for profiling	Points$^{\#}$ of interest	Success Rate
9	4000	414	**100%**
10	4000	180	**100%**
11	4000	391	**100%**
12	4000	524	**100%**
13	4000	315	**100%**
14	4000	508	**100%**
15	4000	370	**100%**
16	4000	416	**100%**
17	4000	409	**100%**
18	4000	367	**100%**
19	4000	513	**100%**
20	4000	70	**100%**

First, we use CPA to detect the leakages for 11-th, 17-th, 18-th and 19-th bit of y with 4,500 traces gathered in each group. In practical attacking, we use Template Attack (TA, one of profiling attacks) to recover 11-th, 17-th, 18-th and 19-th bit in y. Simply, we use lots of traces (no more than 4500) to profile for $bit = 0$ and $bit = 1$, then use other 406 traces to verify whether we can achieve the 100% Success Rate using one trace to recover these four bits. The attack results are shown in Table 13.

Table 13. The results of TA on randomness generation aiming to recover any bit in y.

i-th bit	trace$^{\#}$ for profiling	Points$^{\#}$ of Interesting	Success Rate
11	4500	75	**100%**
17	4500	31	**100%**
18	4500	49	**100%**
19	4500	42	**100%**

F.3 A Blind SCA on Polynomial Addition

In this section, we describe how to get 1 bit information of y without a profiling step. A software platform is used to implement the sensitive operation related to y (i.e. $\mathbf{z} = \mathbf{y} + \mathbf{sc}$). The implementation environment, acquisition equipment and parameters setting are the same as those in Section F.1.

First, we need to make the some assumptions which we claim is very usual and realistic. We assume that the adversary knows the approximate Points of Interesting (PoI) related to randomness y (since the characters of leakages can be observed from each trace). Next, use a window to capture a part of traces which include PoI, and launch a blind side-channel attack. Specifically, cluster these parts into 2 groups (by K-means in this experiment), then one group maps to $bit_x(y) = 0$, and the other group maps to $bit_x(y) = 1$, where $bit_x(y)$ denotes the x-th bit of y. And the adversary can guess arbitrary since it dose not affect this attack (guess again if guessed wrong).

In practical attacking, we use blind SCA (a non-profiling attack) to recover 8-th bit and 11-th bit in y. Simply, we collected 2,000 traces to attack, and the success rate is significantly affected by the size of window. The attack results are shown in Table 14.

Table 14. The results of blind SCA aiming to recover 8-th bit and 11-th bit in y.

bit	size of window	Success Rate
8-th	50	**97.05%**
	100	**97.10%**
	150	**96.15%**
	200	61.00%
11-th	50	**86.75%**
	100	**86.90%**
	150	**86.95%**
	200	53.05%

As shown in Table 14, while 2000 traces are used to cluster and attack, we can recover the 8-th bit and 11-th bit with probabilities close to 1, which demonstrate that the randomness can be recovered without profiling SCAs.