One Bit to Rule Them All – Imperfect Randomness Harms Lattice Signatures

Simon Damm , Nicolai Kraus , Alexander May , Julian Nowakowski , and Jonas Thietke

Ruhr-University Bochum
{simon.damm,nicolai.kraus,alex.may,julian.nowakowski,jonas.thietke}@rub.de

Abstract. The Fiat-Shamir transform is one of the most widely applied methods for secure signature construction. Fiat-Shamir starts with an interactive zero-knowledge identification protocol and transforms this via a hash function into a non-interactive signature. The protocol's zero-knowledge property ensures that a signature does not leak information on its secret key $\mathbf{s}$, which is achieved by blinding $\mathbf{s}$ via proper randomness $\mathbf{y}$. Most prominent Fiat-Shamir examples are DSA signatures and the new post-quantum standard Dilithium.

In practice, DSA signatures have experienced fatal attacks via leakage of a few bits of the randomness $\mathbf{y}$ per signature. Similar attacks now emerge for lattice-based signatures, such as Dilithium.

We build on, improve and generalize the pioneering leakage attack on Dilithium by Liu, Zhou, Sun, Wang, Zhang, and Ming. In theory, their original attack can recover a 256-dimensional subkey of Dilithium-II (aka ML-DSA-44) from leakage in a single bit of $\mathbf{y}$ per signature, in any bit position $j \geq 6$. However, the memory requirement of their attack grows exponentially in the bit position j of the leak. As a consequence, if the bit leak is in a high-order position, then their attack is infeasible.

In our improved attack, we introduce a novel transformation, that allows us to get rid of the exponential memory requirement. Thereby, we make the attack feasible for *all* bit positions $j \geq 6$. Furthermore, our novel transformation significantly reduces the number of required signatures in the attack.

The attack applies more generally to all Fiat-Shamir-type lattice-based signatures. For a signature scheme based on module LWE over an ℓ-dimensional module, the attack uses a 1-bit leak per signature to efficiently recover a $\frac{1}{\ell}$-fraction of the secret key. In the ring LWE setting, which can be seen as module LWE with $\ell = 1$, the attack thus recovers the whole key. For Dilithium-II, which uses $\ell = 4$, knowledge of a $\frac{1}{4}$-fraction of the 1024-dimensional secret key lets its security estimate drop significantly from 128 to 84 bits.

1 Introduction

DSA Attacks as a Warning. DSA signatures are one of the most important cornerstones for securing authenticity in our digital society. The origin of DSA

lies in Schnorr's identification protocol [Sch90] that proves knowledge of a secret discrete logarithm s via revealing some value $z = c \cdot s + y \bmod q$, where c is a known challenge and y an unknown randomness. It is easy to see that for y chosen uniformly at random from $\mathbb{Z}_q$, z is also uniformly random over $\mathbb{Z}_q$. Hence, the value of $c \cdot s$ is information theoretically hidden, thereby perfectly blinding the secret s.

On the attack side, it was soon realized [HGS01, NS02, NS03] that the Boneh-Venkatesan lattice-based algorithm for the *hidden number problem* [BV96] can be utilized to tackle DSA signatures via leakage of y's bits. While the original theoretical bound required $\mathcal{O}(\sqrt{\log q})$ leaked bits of y per signature, this was quickly improved to a few bits of y in practical DSA settings. Ever since then, the cryptographic community has been on a chase for further reducing the required amount of leaked bits per signature [Aka09, DHMP13, ANT+20, AH21, XSWH22, HR23].

A series of devastating real-world attacks [HR07, Rya18, MBA+21, HR23, CVE] demonstrated that *randomness leakage* is not only a theoretical threat. Interestingly, these real-world attacks usually do not require invasive side-channel techniques. Instead, they often simply exploit a slight bias in the choice of y, e.g., when some bits of y are (unintentionally) set to a fixed value. Moreover, the Dual EC disaster [CNE+14] showed that randomness selection may also be biased maliciously. This warns us to carefully secure Fiat-Shamir based post quantum signatures against similar randomness leakage attacks.

History of Code- and Lattice-based Signatures. When looking at post-quantum cryptography in general, the construction of efficient and secure signatures has been a more delicate process than the construction of encryption. While initial constructions for encryption from codes [McE78] and lattices [GGH97, HPS98] basically resisted cryptanalytic efforts and just underwent some modernizations [ABC+22, HRSS17], the story of their signature counterparts [CFS01, HPS01, HHP+03] has seen a series of breaks and improvements.

For codes, the well-known McEliece-type CFS signature [CFS01] suffered from slow signing, initial parameters were broken by a Generalized Birthday attack due to Bleichenbacher (see [FS09]), and later a key distinguishing attack was found [FGUO+13].

For lattices, the NTRU NSS scheme [HPS01] and its successor NTRUSign [HHP+03] have faced effective cryptanalytic attacks [GJSS01, GS02]. Nguyen and Regev [NR09] showed that the inherently leaked information of GGH and NTRU signatures [GGH97, HHP+03] can be exploited to recover the secret key via gradient descent on a multivariate optimization problem. The result of [NR09] already demonstrated that even a small leakage is a serious threat to lattice signing schemes, resulting in full key recovery.

In two breakthrough results on the constructive side, Gentry, Peikert, Vaikuntanathan [GPV08] and Lyubashevsky [Lyu09] demonstrated how to build lattice-based signatures, provably without secret key leakage. While [GPV08] utilizes the hash-and-sign paradigm, Lyubashevsky [Lyu09] uses the Fiat-Shamir transform.

Lyubashevsky provides a method called *Fiat-Shamir with Aborts* for achieving a zero-knowledge proof of an LWE secret key $\mathbf{s}$, heavily inspired by Schnorr's protocol [Sch90]. The novel post-quantum standard Dilithium is essentially a highly optimized variant of Lyubashevsky's signature scheme.

Fiat-Shamir with Aborts. Let us dive a little deeper into the details of Lyubashevsky's Fiat-Shamir with Aborts [Lyu09, DFPS23] in various LWE settings. Let $R = \mathbb{Z}[X]/(X^n + 1)$, and let $\mathbf{s} \in R^\ell$ be an LWE secret key having ℓn (small) polynomial coefficients. The reader should think of ℓn as the security parameter. For a challenge $c \in R$ derived from a hashed message, Lyubashevsky's scheme blinds the value $c\mathbf{s}$ via some randomness $\mathbf{y} \in R^\ell$ as $\mathbf{z} := c\mathbf{s} + \mathbf{y}$, analogous to [Sch90].

Importantly, while Schnorr's scheme is defined over the *finite* ring $\mathbb{Z}_q$, Lyubashevsky uses the *infinite* ring $R = \mathbb{Z}[X]/(X^n + 1)$. Using an infinite ring comes with the disadvantage that $\mathbf{z}$ can no longer be uniformly random over R. Thereby, $\mathbf{z}$ may leak information on $\mathbf{s}$. To prevent this, Lyubashevsky introduces a clever rejection sampling technique, which reruns the underlying identification protocol, until the resulting $\mathbf{z}$ becomes independent of $\mathbf{s}$. However, this zero-knowledge argument crucially requires that $\mathbf{y}$ is chosen *perfectly secret and uniformly at random*.

Randomness Bit Leakage Model. For simplicity, in the remainder, we call $(c, \mathbf{z})$ a signature, although a Lyubashevsky signature contains more information (that we do not use in our attack).

Notice that the randomness $\mathbf{y} \in R^\ell$ is represented as ℓn polynomial coefficients, denoted $y_1, \ldots, y_{\ell n} \in \mathbb{Z}$. As a running example, let us use Dilithium-II parameters (aka ML-DSA-44) with

$$\ell n = 1024, \text{ and } y_1, \ldots, y_{\ell n} \in (-2^{17}, 2^{17}),$$

resulting in a randomness $\mathbf{y}$ having $1024 \cdot 18 = 18{,}432$ bits. Let us write an 18-bit polynomial coefficient y_i in binary representation as

$$y_i = \sum_{j=0}^{17} y_{i,j} 2^j \text{ with } y_{i,j} \in \{0, 1\},$$

e.g., in the standard *binary two's complement* form. Out of the 18,432 bits for representing $\mathbf{y}$, we assume leakage of *a single fixed bit* $y_{i,j} \in \{0, 1\}$ in a position $j \geq 6$. Our goal is to reconstruct $\mathbf{s}$ from many *leaky signatures* $(c, \mathbf{z}, y_{i,j})$, for $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ (with proper rejection sampling) and leakage bit $y_{i,j}$ of y.

Integer LWE (ILWE). It was observed in [HM17] that *Integer LWE* (ILWE) – i.e., an LWE instance without modular reduction, such as Galbraith's binary matrix LWE [Gal13] – leads to efficient cryptanalysis using methods from linear optimization. In [EFGT17], Espitau, Fouque, Gérard and Tibouchi described a

side-channel attack on BLISS signatures that via leakage also leads to ILWE equations. Afterwards, the ILWE problem was systematically studied by Bootle, Delaplace, Espitau, Fouque and Tibouchi [BDE+18], who showed that ILWE can in general be solved by linear regression. More precisely, given an LWE instance $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$ over the integers, one can use linear regression to efficiently compute an approximation $\hat{\mathbf{s}}$ of $\mathbf{s}$ over the reals. The authors of [BDE+18] showed that, if the LWE instance has enough *samples* (i.e., $\mathbf{A}$ provides sufficiently many rows), then rounding $\hat{\mathbf{s}}$ coordinate-wise reveals $\mathbf{s}$.

Liu-Zhou-Sun-Wang-Zhang-Ming Attack [LZS+20]. By construction, a Dilithium signature $(c, \mathbf{z})$ already defines an ILWE instance $\mathbf{z} = c\mathbf{s} + \mathbf{y}$, where $\mathbf{y}$ plays the role of an LWE error. However, by Lyubashevsky's zero knowledge result [Lyu09] those equations perfectly protect $\mathbf{s}$, resulting in an unsolvable ILWE instance.

The situation changes if we leak some bit $y_{i,j}$ of $\mathbf{y}$ for many signatures $(c, \mathbf{z})$, as first observed in the attack of Liu, Zhou, Sun, Wang, Zhang and Ming [LZS+20]. For Dilithium-II, using knowledge of a single leaked bit $y_{i,j}$ in position $j \geq 6$, the work of [LZS+20] shows how to transform the unsolvable ILWE instance $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ into a solvable ILWE instance, whose solution is a 256-dimensional subkey $\bar{\mathbf{s}}$ of Dilithium-II's 1024-dimensional secret $\mathbf{s}$. The authors of [LZS+20] apply the ILWE framework of [BDE+18] to solve via linear regression for the subkey $\bar{\mathbf{s}}$. For bit position $j = 6$, the regression successfully recovers all 256 coordinates of $\bar{\mathbf{s}}$ using roughly half a million signatures, that in turn lead to half a million ILWE relations. Given the efficiency of linear regression, carrying out such an attack can be done in less than 1 minute for recovery of $\bar{\mathbf{s}}$.

However, for bit positions $j > 6$, the attack is significantly less efficient. The reason is that, in the [LZS+20] attack, increasing j by 1 increases the number of required signatures by a factor roughly 4. This exponential increase in *signatures and ILWE relations* quickly makes the [LZS+20] attack impractical. As an example, if the bit leak is in the most significant bit $j = 17$, the attack would require solving an ILWE instance with roughly 2^{41} relations, which – due to the large memory consumption – is currently out of reach.

For $j = 6$, the work of Qiao, Liu, Zhou, Ming, Jin and Li [QLZ+22] demonstrated the real-world relevance of the [LZS+20] attack, by realizing randomness bit leakage via a Public Template Attack on a masked Dilithium implementation.

1.1 Our Results

Our starting point is the [LZS+20] attack within the ILWE framework of [BDE+18]. However, we strongly deviate from the original description of [LZS+20].

Understanding Leakage and Zero-Knowledge. The original analysis shows that leakage in $\mathbf{y}$ allows generation of new ILWE relations that, seemingly *by chance*, fall into a regime where linear regression can recover the subkey $\bar{\mathbf{s}}$. A

significant drawback of this approach is, however, that it can not explain *why* leakage in $\mathbf{y}$ actually undermines the zero-knowledge property of the signature scheme. This makes building upon the attack quite challenging. Therefore, we develop a completely new approach towards the attack. We formally analyze the distribution of leaky LWE signatures in detail, allowing us to fully explain how leakage in $\mathbf{y}$ undermines zero-knowledge. This, in turn, allows us to significantly improve the attack in various ways.

Informative Relations. Our novel analysis shows that for increasing bit positions j, most of the resulting ILWE relations actually do not provide *any* information on $\mathbf{s}$. We call such relations *zero-knowledge*. Those relations that actually provide information are called *informative*.

Using again Dilithium-II as an example, [LZS+20] feeds for $j > 6$ a mixture of *many* zero-knowledge and *few* informative relations as input to linear regression. The use of zero-knowledge relations does not only unnecessarily slow down linear regression, but also requires a larger amount of relations in total, since the zero-knowledge relations dilute the input data. If we instead feed only informative relations as input to linear regression, then the amount of required signatures already drops significantly. As a numerical example, while the [LZS+20] attack requires for $j = 9$ around 30 million signatures, we improve to around 2 million.

In the original attack of [LZS+20], the number of required signatures roughly grows by a factor of 4 for every increase in bit position j. In ours, the growth per bit position drops to roughly a factor of 2. Notice that such a (still) exponential growth in j is inherent. The reason is that the amount of zero-knowledge relations roughly doubles per increase in j.

Constant Memory. While obtaining a large number of signatures is generally not problematic in practice, the main bottleneck in [LZS+20] lies in the fact that all the collected signatures must be stored and subsequently used as input for linear regression.

We significantly improve on this state of affairs, by processing the signatures *as a stream*, and storing only those that yield informative relations. After that, we employ a novel transformation to our informative relations. We formally prove that our transformation makes the error term in the underlying ILWE instance independent of the bit position j. Thereby, we require for any bit position j the same number of informative relations for linear regression to succeed. In particular, we achieve *constant memory* and thus make the attack feasible for all bit positions $j \geq 6$.

Power of the Ring. In a Lyubashevsky-type signature scheme based on module LWE over an ℓ-dimensional module, each signature coefficient is a function of only a subkey, an $\frac{1}{\ell}$-fraction of the complete secret key $\mathbf{s}$. In the ring LWE setting with $\ell = 1$, the subkey is in fact the whole secret key $\mathbf{s}$. For Dilithium-II with $\ell = 4$ each signature coefficient depends on a 256-coordinate subkey, a

$\frac{1}{4}$-fraction of the full 1024-coordinate $\mathbf{s}$. Thus, assuming only a single bit leak our attack can naturally only recover the corresponding subkey on which the signature coefficient depends. If multiple bits are leaked – one for each of the ℓ rings in module LWE – the attack recovers all subkeys, and thus the complete secret key.

However, throughout the paper we assume only a single bit leak. Thus, our main goal is efficient subkey recovery. For completeness, we also briefly analyze the complexity of recovering the whole key from a $\frac{1}{\ell}$-fraction subkey using standard lattice based methods [DDGR20].

Analysis of Required Relations. The work of [BDE+18] describes a very general framework for solving ILWE via linear regression. By an application of the [BDE+18] framework, [LZS+20] achieve a lower bound for the number of required linear relations to successfully recover a secret key $\mathbf{s}$ via rounding that is quite inaccurate in practice. We provide an improved bound, by fine-tuning the analysis to our attack setting, that accurately matches our experimental results.

Organization of the paper. In Section 2, we recall Lyubashevsky's identification protocol, that together with the Fiat-Shamir transform results in Lyubashevsky's signature scheme. After defining our attack model with leaky LWE signatures in Section 3, we describe our improvements to the [LZS+20] attack in Section 4. Correctness and run time of our attack are analyzed in Section 5, and experimental results are provided in Section 6. Eventually, we discuss in Section 7 methods to save on the required amount of signatures by combining linear regression with lattice reduction.

2 Lyubashevsky ID protocol, Fiat-Shamir with Aborts

In this section, we recall Lyubashevsky's identification protocol [Lyu09, Lyu24] (ID protocol), and introduce useful notations.

Notations. Lyubashevsky's ID protocol is defined over the *cyclotomic ring* $R := \mathbb{Z}[X]/(X^n+1)$, where n is a power of two. Some operations are performed over the ring $R_q := \mathbb{Z}_q[X]/(X^n+1)$, for some prime q. Every ring element $r \in R$ is represented by a degree-$(n-1)$ polynomial $r = \sum_{i=0}^{n-1} r_i X^i$, where $r_i \in \mathbb{Z}$. For a ring element $r \in R$, we define $[r]_q := r \mod q$, i.e., applying $[\cdot]_q$ to r reduces the coefficients r_i modulo q. We extend $[\cdot]_q$ to vectors $\mathbf{v} \in R^\ell$ by applying it coordinate-wise. We stress that throughout our paper, *all* operations are performed over R and $\mathbb{Z} \subseteq R$, unless we explicitly indicate modular reduction by the $[\cdot]_m$ operator, for some modulus $m \in \mathbb{N}$.

For $\gamma \in \mathbb{N}$, we write $[\pm\gamma]^n \subset R$ to denote the set of all ring elements $r \in R$ with coefficients $|r_i| \leq \gamma$. If $n = 1$, then we drop the n-superscript from $[\pm\gamma]^n$. Moreover, for $0 \leq \tau \leq n$, we define $[\pm 1]^n_\tau \subseteq [\pm 1]^n \subset R$ as the set of all ring elements $r \in [\pm 1]^n \subset R$ having exactly τ non-zero coefficients.

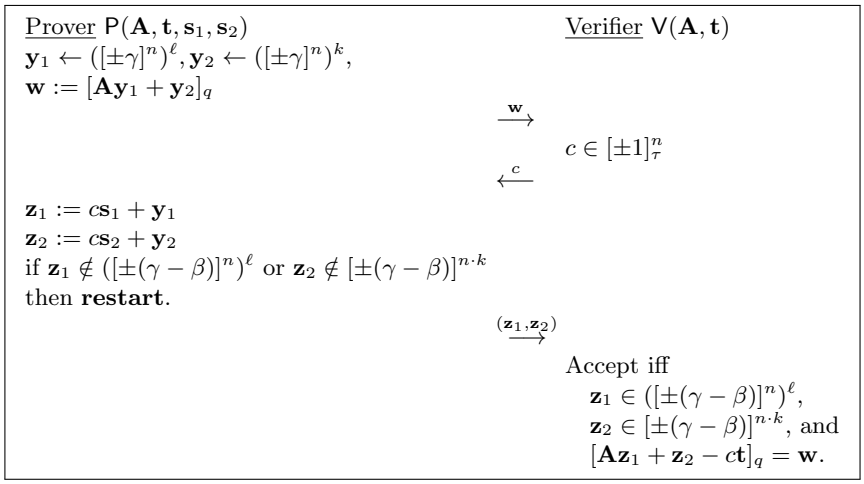

Fig. 1. Lyubashevsky's ID protocol.

Lyubashevsky's protocol. The ID protocol is parametrized by

- the ring dimension n,
- the modulus q,
- the LWE-secret dimension ℓ,
- the LWE-error dimension k,
- the LWE-distribution width η,
- the commitment-distribution width γ,
- the randomness-distribution width γ, and
- the challenge weight τ.

For notational convenience, we further define $\beta := \eta \cdot \tau$.

Public keys are of the form $(\mathbf{A}, \mathbf{t}) \in R_q^{k \times \ell} \times R_q^k$, where $\mathbf{A} \in_R R_q^{k \times \ell}$ and $\mathbf{t} = [\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2]_q$, for some $\mathbf{s}_1 \in_R ([\pm\eta]^n)^\ell$ and $\mathbf{s}_2 \in_R ([\pm\eta]^n)^k$. (Throughout the paper, for a set A, we denote by $a \in_R A$ that a is sampled uniformly at random from A.) The secret key corresponding to $(\mathbf{A}, \mathbf{t})$ is $(\mathbf{s}_1, \mathbf{s}_2)$. Hence, a key pair defines a (module) LWE instance. The goal of the ID protocol is to create a zero-knowledge proof of knowledge of the secret key.

Following the notation of Lyubashevsky's recent survey [Lyu24, Figure 5], we formally describe the ID Protocol in Figure 1. It follows the usual three-step ID structure of the prover P sending a *commitment* $\mathbf{w}$ (depending on some randomness $(\mathbf{y}_1, \mathbf{y}_2)$), the verifier V sending a *challenge* c, and the prover sending a *response* $(\mathbf{z}_1, \mathbf{z}_2)$.

Notably, the prover P restarts the protocol, if the coefficients of $\mathbf{z}_1$ and $\mathbf{z}_2$ do not fall into the range $[\pm(\gamma - \beta)]$. This *rejection sampling* of *Fiat-Shamir with Aborts* is crucial for zero-knowledge. To information-theoretically hide the secret key, the coefficients of the response have to be uniformly distributed over the range $[\pm(\gamma - \beta)]$.

Typical parameters for the ID protocol are shown in Table 1. The *ML-DSA* columns show the three standardized Dilithium parameter sets. We provide an additional parameter set, labelled *Ring-LWE-1024*, which is supposed to have the same security level as the ML-DSA-44 parameters. Indeed, in ML-DSA-44 and Ring-LWE-1024, we have $k \cdot n = \ell \cdot n = 1024$, and all remaining parameters are identical. The complexity of all known attacks does not depend on the values of n and ℓ themselves, but only on the product $\ell \cdot n$. However, as we will discuss in the subsequent sections, in the presence of leakage, the values of ℓ and n are very important for security. That is, the smaller ℓ, the more dangerous leakage becomes.

Table 1. Various parameter sets for Lyubashevsky's ID protocol.

	ML-DSA-44	Ring-LWE-1024	ML-DSA-65	ML-DSA-87
n	256	1024	256	256
k	4	1	6	8
ℓ	4	1	5	7
q	8380417	8380417	8380417	8380417
η	2	2	4	2
γ	2^{17}	2^{17}	2^{19}	2^{19}
τ	39	39	49	60
β	78	78	196	120

Differences with Dilithium. For efficiency purposes, Dilithium uses an optimized variant of the ID protocol. Most importantly, it incorporates various clever techniques, that allow to drop $\mathbf{z}_2$ from the response, and to drop the low bits of $\mathbf{t}$ from the public key. We refer the reader to Lyubashevsky's survey [Lyu24, Sections 5.4 and 5.5] for an in-depth explanation of these optimizations.

For ease of notation, throughout the paper we mostly consider the non-optimized version of the ID protocol, as depicted in Figure 1. We stress, however, that all our results apply to the optimized variant used in Dilithium as well. In particular, dropping $\mathbf{z}_2$ from the response does not affect our attack, since we attack only the $\mathbf{z}_1$-component of the response. Dropping the low bits of $\mathbf{t}$ from the public key also has no effect on our attack, since they can efficiently be recovered via linear programming [OVCG24].

3 Attack Model

We now formally describe our attack model. Consider a Lyubashevsky-type signature scheme based on the ID protocol from Figure 1. In a nutshell, we assume that there is a flaw in the random number generator for sampling $\mathbf{y}_1 \in R^\ell = (\mathbb{Z}[X]/(X^n+1))^\ell$. More precisely, we consider the following scenario: Implementations typically identify $\mathbf{y}_1$ with its $(n \cdot \ell)$-dimensional coefficient vector over

$\mathbb{Z}^{n\cdot\ell}$. The entries of the coefficient vector are stored in binary two's complement. We assume that *one* fixed bit in this binary two's complement is revealed to the attacker. One possible scenario might be, e.g., that in every signature this bit is stuck at 0. Of course, more general scenarios are also possible.

Binary Two's Complement. For a word width w, the binary two's complement stores signed integers x with $-2^{w-1} \leq x < 2^{w-1}$ as a binary string $(x_{w-1}, x_{w-2}, \ldots, x_1, x_0) \in \{0,1\}^w$, such that

$$x = \left[\sum_{j=0}^{w-1} x_j 2^j \right]_{2^w},$$

where $[\cdot]_{2^w}$ denotes the modulo-2^w operator, that maps any $a \in \mathbb{Z}$ to the unique centered around 0 value $[a]_{2^w}$ with $[a]_{2^w} \equiv a \mod 2^w$ and $-2^{w-1} \leq [a]_{2^w} < 2^{w-1}$. See Table 2 for an example of the binary two's complement representations with word width $w = 3$. For a given j, $0 \leq j < w$, we call x_j the *j-th bit in the binary two's complement representation of x*.

Table 2. Binary two's complement representations with word width $w = 3$.

integer x	3	2	1	0	-1	-2	-3	-4
binary two's complement	011	010	001	000	111	110	101	100

Formalizing our Problem. In Lyubashevsky's signature scheme, a signature contains (among other values) a ring element $c \in R$ and a vector $\mathbf{z}_1 \in R^\ell$. Recall that $\mathbf{z}_1$ is computed as

$$\mathbf{z}_1 = c\mathbf{s}_1 + \mathbf{y}_1,$$

where $\mathbf{s}_1$ is the LWE secret, coming from the secret key $(\mathbf{s}_1, \mathbf{s}_2) \in R^\ell \times R^k$. It is easy to see that each of the $n \cdot \ell$ coefficients of $\mathbf{z}_1$ yields one linear relation in the LWE secret $\mathbf{s}_1$. Importantly, since $\mathbf{s}_1 = (s_{1,1}, s_{1,2}, \ldots, s_{1,\ell})$ is an ℓ-dimensional vector over the ring R, the first n coefficients of $\mathbf{z}_1$ depend only on the first component $s_{1,1}$, the next n-coefficients depend only on the second component $s_{1,2}$, and so forth. Hence, in our attack model, we are tasked with solving the following problem:

Definition 1 (Leaky-Signature-LWE). *Fix some public parameters j, τ, η, γ, and define $\beta := \eta \cdot \tau$. Let $(\mathbf{A}, \mathbf{t}) \in R_q^{k\times\ell} \times R_q^k$ be an LWE public key with corresponding secret key $(\mathbf{s}_1, \mathbf{s}_2) \in ([\pm\eta]^n)^\ell \times ([\pm\eta]^n)^k$. Let $\mathbf{x} \in \mathbb{Z}^n$ be the coefficient vector of one component of $\mathbf{s}_1$. In the* Leaky-Signature-LWE *problem one is given the public key $(\mathbf{A}, \mathbf{t})$, along with arbitrarily many relations of the form $(\mathbf{c}, z, y_j) \in [\pm 1]_\tau^n \times \mathbb{Z} \times \{0,1\}$, where z*

$$z = \langle \mathbf{c}, \mathbf{x} \rangle + y,$$

for some $y \in [\pm\gamma]$, such that

1. *z follows the uniform distribution over $[\pm(\gamma - \beta)]$, and*
2. *y_j is the j-th bit in the binary two's complement representation of y.*

The goal is to recover the LWE secret $\mathbf{s}_1$.

4 An Improved Algorithm for Leaky-Signature-LWE

Let us now describe our algorithm for solving the Leaky-Signature-LWE problem from Definition 1. In their seminal work [LZS+20], Liu, Zhou, Sun, Wang, Zhang, and Ming, already gave an algorithm for recovering the partial LWE secret $\mathbf{x}$ in the Leaky-Signature-LWE problem. However, as discussed in the introduction, for high-order bit leakage positions j their algorithm becomes impractical.

In this section, we present a novel and significantly simplified view on their algorithm. By that, we obtain new insights, that allow us to make the algorithm practical for all j.

4.1 Breaking Zero-Knowledge via y_j

We begin by rephrasing the main idea behind the attack of [LZS+20]. We strongly deviate from the original description of [LZS+20], and instead follow a more information theoretic approach. We believe that this helps to gain a deeper understanding of the attack.

In the Leaky-Signature-LWE problem, we obtain relations $(\mathbf{c}, z, y_j)$, where z is defined as

$$z = \langle \mathbf{c}, \mathbf{x} \rangle + y$$

and y_j is the j-th bit in the binary two's complement representation of y. Recall that z is a coefficient of the value $\mathbf{z}_1$ computed in the ID protocol from Figure 1. By rejection sampling, z follows the uniform distribution over $[\pm(\gamma - \beta)]$, for some parameters β and γ. As a reminder, we note that γ satisfies $|y| \leq \gamma$, and that β satisfies $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta$.[1]

In his original work [Lyu09], Lyubashevsky essentially showed that conditioning the distribution of z to the range $[\pm(\gamma-\beta)]$ makes it independent of the secret key. Thereby, the protocol becomes zero-knowledge. For attack purposes, it is convenient to not only consider the range, where the protocol is zero-knowledge, but to specifically consider, where it is *not*. To this end, we need the following lemma.

Lemma 2. *Let $\langle \mathbf{c}, \mathbf{x} \rangle \in \mathbb{Z}$ be a random inner product drawn from some probability distribution. Suppose we sample $y \in \mathbb{Z}$ uniformly at random from $[\pm\gamma]$, independently from $\langle \mathbf{c}, \mathbf{x} \rangle$. Then for $z := \langle \mathbf{c}, \mathbf{x} \rangle + y$ and any $x \in \mathbb{Z}$ it holds that*

$$\Pr[z = x] \propto \Pr[x - \gamma \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma],$$

i.e., the probabilities $\Pr[z = x]$ and $\Pr[x - \gamma \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma]$ are proportional.

[1] By Definition 1, we have $\beta := \eta \cdot \tau$, for some parameters η and τ, such that $\mathbf{x} \in [\pm\eta]^n$ and $\mathbf{c} \in [\pm 1]^n_\tau$. Hence, $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \eta \cdot \tau = \beta$.

Proof. By definition of z, we have

$$\begin{aligned}\Pr[z = x] &= \Pr[y = x - \langle \mathbf{c}, \mathbf{x} \rangle] \\ &= \sum_{i=-\infty}^{\infty} \Pr[y = x - i \mid \langle \mathbf{c}, \mathbf{x} \rangle = i] \cdot \Pr[\langle \mathbf{c}, \mathbf{x} \rangle = i].\end{aligned}$$

Since y is uniformly random over $[\pm\gamma]$ and independent from $\langle \mathbf{c}, \mathbf{x} \rangle$, the above becomes

$$\begin{aligned}\Pr[z = x] &= (2\gamma + 1)^{-1} \sum_{i=x-\gamma}^{x+\gamma} \Pr[\langle \mathbf{c}, \mathbf{x} \rangle = i] \\ &= (2\gamma + 1)^{-1} \cdot \Pr[x - \gamma \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma],\end{aligned}$$

which concludes the proof. □

In the ID protocol from Figure 1, any coefficient z of $\mathbf{z}_1$ is computed as $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$. Before applying rejection sampling, y is uniformly random over $[\pm\gamma]$. Hence, from Lemma 2 and the fact that $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta$, it follows that the distribution of z before rejection sampling is

$$\Pr[z = x] = \alpha \cdot \begin{cases} 1, & \text{if } -\gamma + \beta \leq x \leq \gamma - \beta, \\ \Pr[\langle \mathbf{c}, \mathbf{x} \rangle \geq x - \gamma], & \text{if } \gamma - \beta < x \leq \gamma + \beta, \\ \Pr[\langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma], & \text{if } -\gamma - \beta \leq x < -\gamma + \beta, \\ 0, & \text{else,} \end{cases} \tag{1}$$

for some properly chosen scaling factor α.

By the central limit theorem, the inner product $\langle \mathbf{c}, \mathbf{x} \rangle$ is close to a Gaussian distribution. (Since $\mathbf{c} \in [\pm 1]_\tau^n$ and $\mathbf{x} \in [\pm\eta]$, the inner product $\langle \mathbf{c}, \mathbf{x} \rangle$ is a random sum of τ uniformly random integers from $[\pm\eta]$.) Thus, we may approximate the probability $\Pr[\langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma]$ in Equation (1) by an (appropriately parameterized) cumulative distribution function $\Phi(x + \gamma)$ of the Gaussian distribution. Analogously, we may approximate $\Pr[\langle \mathbf{c}, \mathbf{x} \rangle \geq x - \gamma]$ by $1 - \Phi(x - \gamma)$. The resulting distribution is shown in Figure 2.

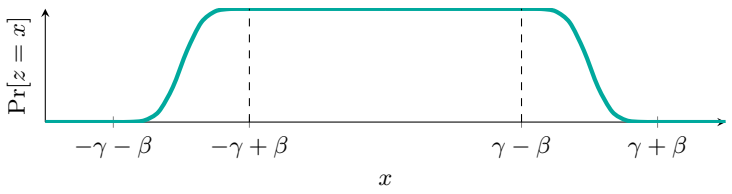

Fig. 2. Distribution of z before rejection sampling.

In Figure 2, the area between $-\gamma + \beta$ and $\gamma - \beta$ (where z follows the uniform distribution) is exactly the area, in which z does not reveal any information

about $\mathbf{s}$. In Lyubashevsky's signature scheme, rejection sampling ensures that we obtain only those z's, that lie inside this exact area.

For *attacking* the signature scheme, however, we would like to have access to z's that lie outside $[\pm(\gamma - \beta)]$. While in our attack scenario, we can not hope to obtain such z's, we show below that by leveraging knowledge of our leaked bit y_j, we can transform our z's, that lie *inside* of $[\pm(\gamma - \beta)]$, into new values $\overline{z}$, that provide as much information about $\mathbf{s}$, as z's *outside* of $[\pm(\gamma - \beta)]$ do.

Leveraging y_j. The authors of [LZS+20] introduce a clever relation extraction technique, that, for all $j \geq \log_2(\beta) + 1$, uses the leaked bit y_j to transform z into a new value $\overline{z}$ of the form

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}. \tag{2}$$

We provide a detailed description of a refined variant of this relation extraction in Section 4.4. For the moment, let us treat the relation extraction in a black box fashion.

Multi-Bit Leakage. One might wonder why we consider for the Leaky-Signature-LWE problem in Definition 1 only a single bit leakage, and how a multi bit leak influences the problem's complexity.

First, single bit leakage is opposed to multi bit leakage a weaker attack model. Thus, obtaining $\mathbf{x}$-recovery from a single bit is a stronger cryptanalytic attack. Second, we show in the following that a multiple bit leak for y does not help an attacker, since only the leak bit y_j in lowest position j matters.

Relation extraction enables us to compute from a relation $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$ the value $\bar{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}$, where $[y]_{2^j}$ represents the (unknown) j low order bits of y. It follows that the computation of $z - \bar{z} = y - [y]_{2^j}$ reveals all most significant bits of y from position j onwards.

Information Extraction Provides Information. Recall that the initial value z lies inside $[\pm(\gamma - \beta)]$, and thus does not provide any information about $\mathbf{x}$. Importantly, the new value $\overline{z}$, on the other hand, *does* reveal information about $\mathbf{x}$. Indeed, after reducing y modulo 2^j, the value $[y]_{2^j}$ is essentially uniformly distributed over $[\pm 2^{j-1}]$.[2] Hence, we may apply Lemma 2 to $\overline{z}$, and conclude that the distribution of $\overline{z}$ is the distribution shown in Figure 3. From Figure 3 it then easily follows that all $\overline{z}$'s outside the range $[\pm(2^{j-1} - \beta)]$ reveal information about $\mathbf{s}$. In fact, those $\overline{z}'s$ reveal exactly as much information, as the initially rejected z's outside the range $[\pm(\gamma - \beta)]$.

Notice that the $\overline{z}$'s now essentially form an instance of the *Integer LWE* problem with solution $\mathbf{x}$ and error drawn uniformly from $[\pm 2^{j-1}]$. As shown

[2] The original analysis of [LZS+20, Section 3, Step 4] falsely assumes that not $[y]_{2^j}$, but $\overline{z}$ follows the uniform distribution modulo 2^j. We note that $[y]_{2^j}$ is not *perfectly* uniformly random over $[\pm 2^{j-1}]$, but rather over $\{-2^{j-1}, -2^{j-1}+1, \ldots, 2^{j-1}-2, 2^{j-1}-1\}$. To simplify notation, we ignore this benign technical detail.

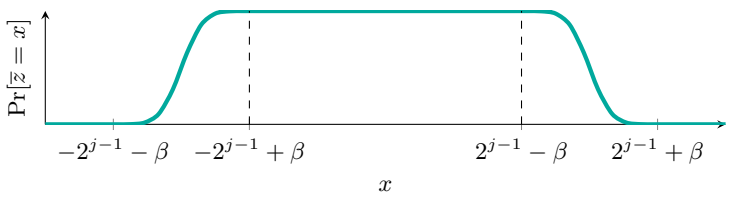

Fig. 3. Distribution of $\overline{z} = \langle \mathbf{c}, \mathbf{s} \rangle + [y]_{2^j}$.

in [LZS+20], when given sufficiently many relations, one can solve this problem in polynomial time via linear regression. We provide in Section 5 improved methods for estimating the number of relations.

Importantly, *polynomial time* does not always mean *feasible* in practice: When trying to solve Integer LWE via linear regression, the required number of relations strongly depends on the error size. Unfortunately, when the error size (and thereby the number of relations) becomes too large, then current state-of-the-art implementations of linear regression become infeasible. In our scenario, the size of the error $[y]_{2^j}$ is directly related to the value of j. That is, the higher j, the larger the error. As a consequence, the original attack of [LZS+20] becomes infeasible for too large j.

As we show in the following Section 4.2, we can make the error size independent of j. This implies that the required amount of relations for linear regression no longer increases with j, thereby making our attack feasible for all leakage bit positions.

4.2 Our Novel Transformation: Achieving Independence of j.

As discussed above, only $\overline{z}'s$ outside the range $[\pm(2^{j-1} - \beta)]$ reveal information about $\mathbf{s}$, let us call relations with these $\overline{z}'s$ *informative*. We call relations with $\overline{z}'s$ inside the range $[\pm(2^{j-1} - \beta)]$ *zero-knowledge*. As a first improvement, we should not feed any *zero-knowledge* relations as input to linear regression.

By including only *informative* relations, we obtain a somewhat odd-looking distribution, as shown in the top half of Figure 4. To alter the shape of this distribution, we propose as our second improvement to transform $\overline{z}$ as

$$\widetilde{z} := \begin{cases} \overline{z} - 2^{j-1} + \beta, & \text{if } \overline{z} \geq 2^{j-1} - \beta, \\ \overline{z} + 2^{j-1} - \beta, & \text{if } \overline{z} \leq -2^{j-1} + \beta. \end{cases} \tag{3}$$

The resulting distribution of $\widetilde{z}$ is shown in the bottom half of Figure 4.

By Equations (2) and (3), we can write our *(transformed) informative* relations as

$$\widetilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y},$$

for some $\widetilde{y}$, that is either $\widetilde{y} = [y]_{2^j} - 2^{j-1} + \beta$ or $\widetilde{y} = [y]_{2^j} + 2^{j-1} - \beta$. As the following theorem shows, $\widetilde{y}$ follows the uniform distribution over $[\pm\beta]$, thereby achieving independence of j.

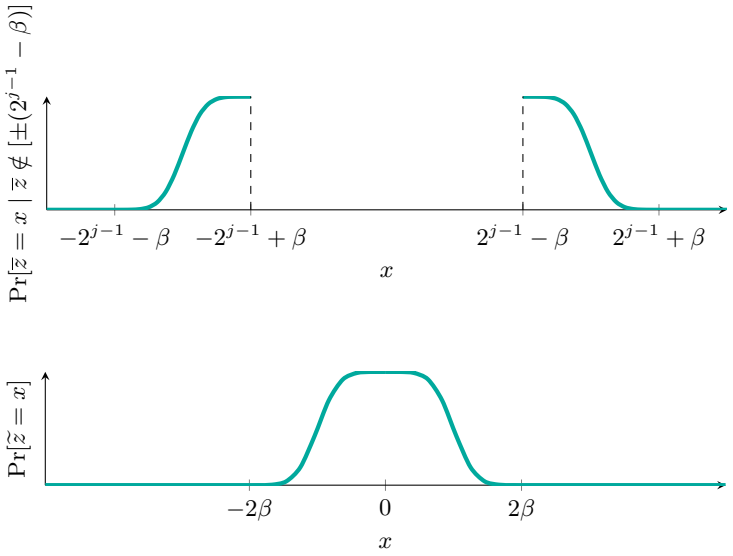

Fig. 4. The top graph shows the distribution of $\overline{z} = \langle \mathbf{c}, \mathbf{s} \rangle + [y]_{2^j}$, conditioned on $\overline{z} \notin [\pm(2^{j-1} - \beta)]$. The bottom graph shows the distribution of $\widetilde{z}$.

Theorem 3. *Let $\mathcal{D}$ be a probability distribution over $\mathbb{Z}$ that produces random inner products $\langle \mathbf{c}, \mathbf{x} \rangle$ such that $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta$ with probability 1. Consider the following probability distribution:*

1. *Sample $\langle \mathbf{c}, \mathbf{x} \rangle$ from $\mathcal{D}$ and $\overline{y}$ uniformly at random from $[\pm 2^{j-1}]$.*
2. *Set $\overline{z} := \langle \mathbf{c}, \mathbf{x} \rangle + \overline{y}$.*
3. *If $-2^{j-1} + \beta < \overline{z} < 2^{j-1} - \beta$, go back to Step 1. Otherwise, compute $\widetilde{z}$ as in Equation (3) and output $\widetilde{y} := \widetilde{z} - \langle \mathbf{c}, \mathbf{x} \rangle$.*

Then $\widetilde{y}$ follows the uniform distribution over $[\pm\beta]$.

Proof. As Figure 4 illustrates, we have

$$\Pr[\widetilde{z} = x] = \alpha \cdot \begin{cases} \Pr[\langle \mathbf{c}, \mathbf{x} \rangle \geq x - \beta], & \text{if } 0 \leq x \leq 2\beta, \\ \Pr[\langle \mathbf{c}, \mathbf{x} \rangle \leq x + \beta], & \text{if } -2\beta \leq x < 0, \\ 0, & \text{else,} \end{cases}$$

for some properly chosen scaling factor α. We can write this more compactly as

$$\Pr[\widetilde{z} = x] \propto \Pr[x - \beta \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \beta].$$

Suppose we sample an integer u uniformly at random from $[\pm\beta]$. Using u, we define $v := \langle \mathbf{c}, \mathbf{s} \rangle + u$. By Lemma 2, we have

$$\Pr[v = x] \propto \Pr[x - \beta \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \beta],$$

for every $x \in \mathbb{Z}$. In particular, $\Pr[\widetilde{z} = x] = \Pr[v = x]$, i.e., $\widetilde{z}$ and v follow the same distribution. It follows that the distribution of the random variables $\widetilde{z} - \langle \mathbf{c}, \mathbf{x} \rangle$ and $v - \langle \mathbf{c}, \mathbf{x} \rangle$ is also identical. Since $\widetilde{z} - \langle \mathbf{c}, \mathbf{x} \rangle = \widetilde{y}$ and $v - \langle \mathbf{c}, \mathbf{x} \rangle = u$, this shows that $\widetilde{y}$ follows the uniform distribution over $[\pm\beta]$, and thus concludes the proof. □

Summarizing the above, by excluding all $\overline{z}$'s inside the range $[\pm(2^{j-1} - \beta)]$ and then transforming the remaining $\overline{z}$'s as in Equation (3), we obtain new Integer LWE relations

$$\widetilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y}, \tag{4}$$

where the error $\widetilde{y}$ follows the uniform distribution over $[\pm\beta]$. This significantly improves over [LZS+20], since now

1. all relations reveal information about $\mathbf{x}$, and
2. the error size decreases from 2^{j-1} to β.[3]

In particular, by our transformation, the error size becomes independent of j, i.e., the bit-index, at which we obtain the leak. By that, the attack becomes feasible also for large j.

4.3 Recovering $\mathbf{s_1}$ from $\mathbf{x}$

If we now apply linear regression to our modified Integer LWE relations from Equation (4), then we efficiently recover the partial key $\mathbf{x} \in \mathbb{Z}^n$. However, to *completely* solve our Leaky-Signature-LWE problem from Definition 1, we have to recover the whole LWE secret $\mathbf{s}_1 \in (\mathbb{Z}[X]/(X^n + 1))^\ell$.

The best known attack strategy for recovering $\mathbf{s}_1$ from $\mathbf{x}$ is based on the *LWE with side information* framework of [DDGR20, DGHK23, MN23]. Here, each of the n-coordinates of $\mathbf{x}$ gives rise to a *perfect hint* on the LWE secret $\mathbf{s}_1$. Given such perfect hints, the framework transforms the lattice problem, that underlies our LWE instance, into a simpler one.

The work of Dachman-Soled, Ducas, Gong and Rossi [DDGR20] provides an estimator, that determines the complexity of the resulting lattice problem within the *Core-SVP model*. In this model, one estimates the smallest *BKZ blocksize* β, at which the BKZ algorithm [Sch87] successfully solves the problem. For a given blocksize β, the bit complexity of BKZ is then estimated as $2^{0.292 \cdot \beta}$.

Dilithium Parameter Sets. We ran the estimator of [DDGR20] on the standardized Dilithium parameter sets (see Table 1), to determine the required BKZ blocksize β for recovering the whole LWE secret key from κ known coordinates (for all $\kappa = 0, 1, 2, \ldots, \ell \cdot n$). The results are shown in Figure 5.

For of our leakage attack, where $\mathbf{x}$ reveals $\kappa = n = 256$ coordinates, we obtain blocksizes $\beta = 287$, $\beta = 469$ and $\beta = 712$ for the parameter sets ML-DSA-44,

[3] As noted above, [LZS+20] require $j \geq \log_2(\beta) + 1$. Thus, going from 2^{j-1} to β is indeed a reduction in error size.

ML-DSA-65 and ML-DSA-87, respectively. The corresponding bit complexities are 84, 136 and 208. We conclude that in all three parameter sets, recovering $\mathbf{s}_1$ from $\mathbf{x}$ still requires significant computational effort. However, knowledge of $\mathbf{x}$ brings security *significantly* below the desired levels of 128, 192 and 256 bits.

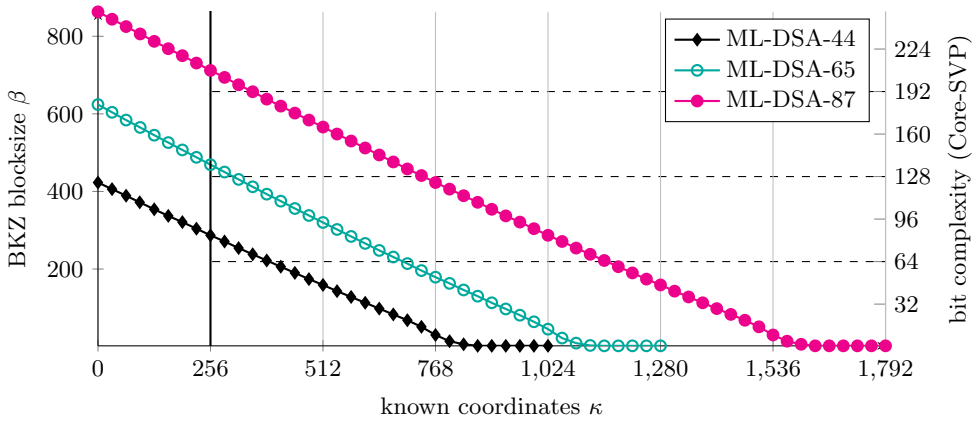

Fig. 5. Required BKZ blocksize to break Dilithium with known coordinates.

Comparison with Ring LWE. The hardness of recovering $\mathbf{s}_1$ from $\mathbf{x}$ strongly depends on the value of ℓ, since the n coordinates of $\mathbf{x} \in \mathbb{Z}^n$ reveal a $\frac{1}{\ell}$-fraction of the LWE secret $\mathbf{s}_1 \in (\mathbb{Z}[X]/(X^n+1))^\ell$. Naturally, the larger ℓ, the harder recovering $\mathbf{s}_1$ from $\mathbf{x}$ becomes.

In the ring LWE setting, where $\ell = 1$, $\mathbf{x}$ fully reveals $\mathbf{s}_1$. Hence, we conclude that instantiating Lyubashevsky's signature scheme with ring LWE makes it particularly vulnerable against the leakage attack.

We like to stress, however, that this only applies to the setting, where leakage occurs in one *fixed* component of $\mathbf{y} \in (\mathbb{Z}[X]/(X^n+1))^\ell$. If we would obtain leakage in all ℓ components of $\mathbf{y}$, then module LWE would be as vulnerable as ring LWE, since our attack would recover all ℓ subkeys of $\mathbf{s}_1$.

4.4 Revisiting the LZS$^+$ Relation Extraction

As discussed in Section 4.1, at the heart of the attack on the Leaky-Signature-LWE problem lies a clever relation extraction from [LZS+20], that, for all $j \geq \log_2(\beta) + 1$, uses the leaked bit y_j to transform z of the form

$$z = \langle \mathbf{c}, \mathbf{x} \rangle + y \in_R [\pm(\gamma - \beta)]$$

into Integer LWE relations $\overline{z}$ of the form

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}. \tag{5}$$

In Section 4.2, we built our transformation on top of this extraction to obtain relations

$$\widetilde{z} = \langle \mathbf{c}, \mathbf{x} \rangle + \widetilde{y},$$

with small error $\widetilde{y} \in_R [\pm\beta]$.

So far, we treated the relation extraction to produce $\overline{z}$ as in Equation (5) in a black box fashion. In this section, we provide a detailed description of a refined variant of this relation extraction. While the original analysis of [LZS+20] required intricate arguments on the bit-level, we can instead resort to simple geometric arguments. Additionally, our variant works for the standard binary two's complement, whereas [LZS+20] considered the less standard *sign-and-magnitude* representation, where one bit is reserved for storing the sign.

Partitioning $\mathbb{Z}_{2^{j+1}}$. Before we can describe our relation extraction, we have to make some simple, yet important, observations about the ring $\mathbb{Z}_{2^{j+1}}$. Throughout this section, we identify $\mathbb{Z}_{2^{j+1}}$ with the set

$$\mathbb{Z}_{2^{j+1}} = \{-2^j, -2^j + 1, \ldots, 2^j - 1\}.$$

We consider two partitions $\mathbb{Z}_{2^{j+1}} = \mathbb{Z}^{\leftarrow}_{2^{j+1}} \cup \mathbb{Z}^{\rightarrow}_{2^{j+1}}$ and $\mathbb{Z}_{2^{j+1}} = \mathbb{Z}^{\uparrow}_{2^{j+1}} \cup \mathbb{Z}^{\downarrow}_{2^{j+1}}$, where

$$\begin{aligned}
\mathbb{Z}^{\leftarrow}_{2^{j+1}} &:= \{-2^j, -2^j + 1, \ldots, -1\},\\
\mathbb{Z}^{\rightarrow}_{2^{j+1}} &:= \mathbb{Z} \setminus \mathbb{Z}^{\leftarrow}_{2^{j+1}},\\
\mathbb{Z}^{\uparrow}_{2^{j+1}} &:= \{-2^{j-1}, -2^{j-1} + 1, \ldots, 2^{j-1} - 1\},\\
\mathbb{Z}^{\downarrow}_{2^{j+1}} &:= \mathbb{Z} \setminus \mathbb{Z}^{\uparrow}_{2^{j+1}}.
\end{aligned}$$

We also define

$$\begin{aligned}
\mathbb{Z}^{\nwarrow}_{2^{j+1}} &:= \mathbb{Z}^{\leftarrow}_{2^{j+1}} \cap \mathbb{Z}^{\uparrow}_{2^{j+1}},\\
\mathbb{Z}^{\nearrow}_{2^{j+1}} &:= \mathbb{Z}^{\rightarrow}_{2^{j+1}} \cap \mathbb{Z}^{\uparrow}_{2^{j+1}},\\
\mathbb{Z}^{\searrow}_{2^{j+1}} &:= \mathbb{Z}^{\rightarrow}_{2^{j+1}} \cap \mathbb{Z}^{\downarrow}_{2^{j+1}},\\
\mathbb{Z}^{\swarrow}_{2^{j+1}} &:= \mathbb{Z}^{\leftarrow}_{2^{j+1}} \cap \mathbb{Z}^{\downarrow}_{2^{j+1}}.
\end{aligned}$$

It is convenient to think of $\mathbb{Z}_{2^{j+1}}$ as a circle, as depicted in Figure 6. As the figure shows, $\mathbb{Z}_{2^{j+1}} = \mathbb{Z}^{\leftarrow}_{2^{j+1}} \cup \mathbb{Z}^{\rightarrow}_{2^{j+1}}$ partitions our circle into a left and a right half. Similarly, $\mathbb{Z}_{2^{j+1}} = \mathbb{Z}^{\uparrow}_{2^{j+1}} \cup \mathbb{Z}^{\downarrow}_{2^{j+1}}$ partitions it into a top and bottom half, and $\mathbb{Z}_{2^{j+1}} = \mathbb{Z}^{\nwarrow}_{2^{j+1}} \cup \mathbb{Z}^{\nearrow}_{2^{j+1}} \cup \mathbb{Z}^{\searrow}_{2^{j+1}} \cup \mathbb{Z}^{\swarrow}_{2^{j+1}}$ partitions it into four quarters.

We now use Figure 6, to prove two technical lemmas.

Lemma 4. *For every $x \in \mathbb{Z}_{2^{j+1}}$, it holds that*

$$[x]_{2^j} = \begin{cases} x, & \text{if } x \in \mathbb{Z}^{\uparrow}_{2^{j+1}},\\ x + 2^j, & \text{if } x \in \mathbb{Z}^{\swarrow}_{2^{j+1}},\\ x - 2^j, & \text{if } x \in \mathbb{Z}^{\searrow}_{2^{j+1}}. \end{cases}$$

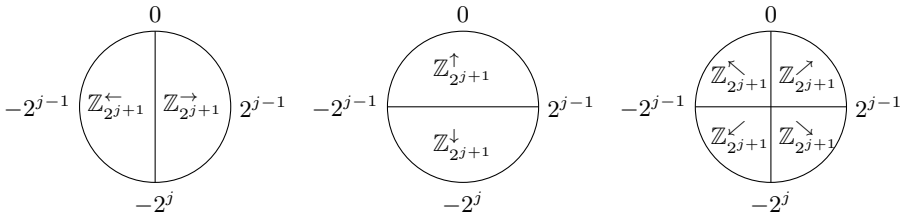

Fig. 6. Intuition for our partitions of $\mathbb{Z}_{2^{j+1}}$.

Proof. Looking at Figure 6, we obtain

$$\begin{aligned}
\mathbb{Z}^{\uparrow}_{2^{j+1}} &= \mathbb{Z}_{2^j}, \\
\mathbb{Z}^{\swarrow}_{2^{j+1}} &= -2^j + \{0, 1, \ldots, 2^{j-1} - 2, 2^{j-1} - 1\}, \\
\mathbb{Z}^{\searrow}_{2^{j+1}} &= 2^j + \{-1, -2, \ldots, -2^{j-1} + 1, -2^{j-1}\},
\end{aligned}$$

which already proves the lemma. □

Lemma 5. *Let $x \in \mathbb{Z}$ be a signed integer with binary two's complement representation $(x_{w-1}, x_{w-2}, \ldots, x_1, x_0) \in \{0,1\}^w$, for some word width w. For any $j < w$ we have*

$$x_j = 1 \iff [x]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}.$$

Proof. By definition of the binary two's complement, we have $x = \left[\sum_{i=0}^{w-1} x_i 2^i\right]_{2^w}$. Together with Figure 6, this proves the lemma. □

With the lemmas above, we are now ready to describe the relation extraction for producing $\overline{z}$ as in Equation (5).

Normal-form Relations. Recall that in the Leaky-Signature-LWE problem, we obtain relations $(\mathbf{c}, z, y_j)$, where z is defined as

$$z = \langle \mathbf{c}, \mathbf{x} \rangle + y,$$

and y_j is the j-th bit in the binary two's complement representation of y. Before doing the actual relation extraction, we apply some pre-processing to our relations. This brings our relations into a special shape, which we call *normal form*. Dealing only with normal form relations allows us to greatly simplify the relation extraction and its analysis.

Our pre-processing transforms a relation $(\mathbf{c}, z, y_j)$, by replacing z and y_j with

$$z^{\uparrow} := \begin{cases} z + 2^{j-1}, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}, \\ z - 2^{j-1}, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}, \end{cases} \tag{6}$$

and

$$b_j := \begin{cases} y_j, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}, \\ y_j \oplus 1, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}, \end{cases} \tag{7}$$

respectively.

Let us explain the effect of our pre-processing. In modulo-2^{j+1} arithmetic, adding 2^{j-1} corresponds to rotating the circles from Figure 6 by 90 degrees clockwise. Similarly, subtracting 2^{j-1} corresponds to rotating the circle by 90 degrees counter clockwise. Hence, Equation (6) ensures that

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}. \tag{8}$$

Let us write $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, for some unknown $y^{\uparrow}$, which (by Equation (6)) is defined as

$$y^{\uparrow} := \begin{cases} y + 2^{j-1}, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}} \\ y - 2^{j-1}, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}} \end{cases}.$$

Assume for a moment that $y^{\uparrow} = y + 2^{j-1}$, i.e, $[z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$. Then by Equation (7), Lemma 5 and the fact that adding 2^{j-1} rotates the circles Figure 6 by 90 degrees clockwise, we have the following equivalence:

$$b_j = 1 \iff y_j = 1 \iff [y]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}} \iff [y^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}.$$

Analogously, if instead $y^{\uparrow} = y - 2^{j-1}$, we have the following equivalence:

$$b_j = 1 \iff y_j = 0 \iff [y]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}} \iff [y^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}.$$

Thus, in any case, it holds that

$$b_j = 1 \iff [y^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}. \tag{9}$$

Relations $(\mathbf{c}, z^{\uparrow}, b_j)$ with $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, for which both Equations (8) and (9) and hold, are called *normal form relations*.

A Simplified LZS$^+$ Relation Extraction. With the normal form relations available after pre-processing, we can now describe our simplified relation extraction for constructing $\overline{z}$. Given a normal form relation $(\mathbf{c}, z^{\uparrow}, b_j)$ with $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, our relation extraction computes

$$\overline{z} := \begin{cases} [z^{\uparrow}]_{2^{j+1}}, & \text{if } b_j = 1, \\ [z^{\uparrow}]_{2^{j+1}} + 2^j, & \text{if } b_j = 0 \text{ and } [z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}, \\ [z^{\uparrow}]_{2^{j+1}} - 2^j, & \text{if } b_j = 0 \text{ and } [z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}. \end{cases} \tag{10}$$

As we show in Theorem 6 below, the resulting $\overline{z}$ has the desired shape. Worth noting, as in the original relation extraction of [LZS$^+$20], we also *crucially* require j to be sufficiently large, such that $j \geq \log_2(\beta) + 1$.

Theorem 6. *Let $(\mathbf{c}, z^{\uparrow}, b_j)$ be a normal form relation with $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, where $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta$. If $j \geq \log_2(\beta) + 1$, then for $\overline{z}$ as defined in Equation (10) it holds that*

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y^{\uparrow}]_{2^j}.$$

Proof. Let us define $u := [z^{\uparrow}]_{2^{j+1}} - \langle \mathbf{c}, \mathbf{x} \rangle$. Then it holds that

$$u \equiv y^{\uparrow} \mod 2^{j+1}, \tag{11}$$

and, in particular, $[u]_{2^j} = [y^{\uparrow}]_{2^j}$. To prove the theorem, we show that

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [u]_{2^j}. \tag{12}$$

Since $(\mathbf{c}, z^{\uparrow}, b_j)$ is a normal form relation, we have by Equation (8) that $[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}$. Let us distinguish the two cases

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\nwarrow}_{2^{j+1}}, \tag{13}$$

and

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\nearrow}_{2^{j+1}}. \tag{14}$$

Suppose we are in the case of Equation (13). Then $[z^{\uparrow}]_{2^{j+1}}$ lies in the upper left quarter of the circles from Figure 7, i.e., $-2^{j-1} \leq [z^{\uparrow}]_{2^{j+1}} < 0$. Moreover, since $j \geq \log_2(\beta) + 1$ and $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta$, we have $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq 2^{j-1}$. It follows that $u = [z^{\uparrow}]_{2^{j+1}} - \langle \mathbf{c}, \mathbf{x} \rangle$ satisfies

$$-2^j \leq u < 2^{j-1}. \tag{15}$$

In other words, u either lies in the upper left quarter, the upper right quarter, or the lower left quarter of the circles from Figure 7, i.e.,

$$u \notin \mathbb{Z}^{\searrow}_{2^{j+1}}. \tag{16}$$

Combining Equations (11) and (15), we obtain $u = [u]_{2^{j+1}} = [y^{\uparrow}]_{2^{j+1}}$. Since $(\mathbf{c}, z^{\uparrow}, b_j)$ is a normal form relation, this implies together with Equation (9) that

$$u \notin \mathbb{Z}^{\downarrow}_{2^{j+1}} \iff u \in \mathbb{Z}^{\uparrow}_{2^{j+1}} \iff [y^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}} \iff b_j = 1.$$

Hence, by Lemma 4 and Equation (16),

$$[u]_{2^j} = \begin{cases} u, & \text{if } b_j = 1, \\ u + 2^j, & \text{else.} \end{cases}$$

If we are instead in the case of Equation (14), one can show completely analogous that

$$[u]_{2^j} = \begin{cases} u, & \text{if } b_j = 1, \\ u - 2^j, & \text{else.} \end{cases}$$

Combining both cases, we obtain

$$[u]_{2^j} = \begin{cases} u, & \text{if } b_j = 1 \\ u + 2^j, & \text{if } b_j = 0 \text{ and } [z^\uparrow]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}, \\ u - 2^j, & \text{if } b_j = 0 \text{ and } [z^\uparrow]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}. \end{cases}$$

Together with Equation (10) and the fact that $[z^\uparrow]_{2^{j+1}} = \langle \mathbf{c}, \mathbf{x} \rangle + u$, this implies Equation (12) and thus concludes the proof of the theorem. □

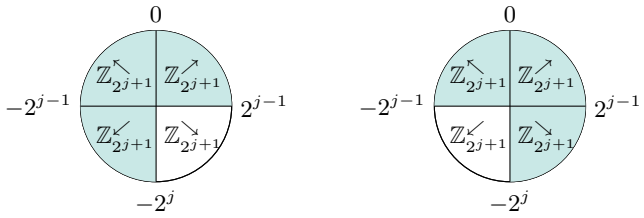

Fig. 7. Intuition for the proof of Theorem 6. The colored area is the range for $u := [z^\uparrow]_{2^{j+1}} - \langle \mathbf{c}, \mathbf{x} \rangle$, depending on whether $[z^\uparrow]_{2^{j+1}} \in \mathbb{Z}^{\nwarrow}_{2^{j+1}}$ (left) or $[z^\uparrow]_{2^{j+1}} \in \mathbb{Z}^{\nearrow}_{2^{j+1}}$ (right).

The Constraint $j \geq \log_2(\beta) + 1$. In the proof of Theorem 6, we crucially require $j \geq \log_2(\beta) + 1$, where β is an upper bound on $|\langle \mathbf{c}, \mathbf{x} \rangle|$. For the ML-DSA-44 parameter set, which has $\beta = 78 \approx 2^{6.3}$ (see Table 1), Theorem 6 thus suggests that our attack requires $j \geq \lceil 6.3 + 1 \rceil = 8$ to work.

However, as we will show in Section 6, our attack works in practice for j as small as $j = 6$. This is due to the fact that, in Dilithium with ML-DSA-44 parameters, the inequality $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq \beta = 78$ is a rather coarse *worst case* bound. Yet, in practice, most $|\langle \mathbf{c}, \mathbf{x} \rangle|$ are significantly smaller than 78: Since $\langle \mathbf{c}, \mathbf{x} \rangle$ is the sum of $\tau = 39$ uniformly random integers from $[\pm\eta] = [\pm 2]$, it follows from the central limit theorem that $\langle \mathbf{c}, \mathbf{x} \rangle$ is close to a Gaussian distribution with mean 0 and variance $\sigma^2 = \frac{(2\eta+1)^2-1}{12} \cdot \tau = 2 \cdot 39 = 78$. (Recall that $\frac{(2\eta+1)^2-1}{12}$ is the variance of the discrete uniform distribution over $[\pm\eta]$.) It is well-known that a Gaussian random variable with variance σ^2 almost never exceeds $3 \cdot \sigma$ in absolute value. Hence, almost all inner products $\langle \mathbf{c}, \mathbf{x} \rangle$ in ML-DSA-44 are bounded by $|\langle \mathbf{c}, \mathbf{x} \rangle| \leq 3 \cdot \sqrt{78} \approx 2^{4.7}$, showing that our attack indeed works for all $j \geq \lceil 4.7 + 1 \rceil = 6$.

For the parameter sets ML-DSA-65 and -87, one can show completely analogously that $\langle \mathbf{c}, \mathbf{x} \rangle$ is close to a Gaussian distribution with standard deviation $\sigma \approx 2^{5.8}$ and $\sigma \approx 2^{5.0}$, respectively. Hence, for these parameter sets, our attack works for all $j \geq 7$ and $j \geq 6$, respectively.

Algorithm 1: Leaky-Signature-LWE Attack

Input : List L relations $(\mathbf{c}, z, y_j)$ with
challenge vector $\mathbf{c} \in [\pm 1]^n_\tau$,
signature coefficient $z \in [\pm(\gamma - \beta)]$,
knowledge of a bit of the randomness $y_j \in [0, 1]$,
leakage index j,
public key $(\mathbf{A}, \mathbf{t})$.

Output: LWE secret $\mathbf{s}_1 \in ([\pm\eta]^n)^\ell$.

1 Initialize empty list $\widetilde{L}$.
2 **for** $(\mathbf{c}, z, y_j) \in L$ **do**
3 **if** $[z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$ **then** ▷ Normal form computation, Eq.(6)&(7).
4 $z^{\uparrow} := z + 2^{j-1}$
5 $b_j := y_j$
6 **else**
7 $z^{\uparrow} := z - 2^{j-1}$
8 $b_j := y_j \oplus 1$
9 **if** $b_j = 1$ **then** ▷ Relation extraction $\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y^{\uparrow}]_{2^j}$, Eq.(10).
10 $\overline{z} := [z^{\uparrow}]_{2^{j+1}}$
11 **else if** $[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$ **then**
12 $\overline{z} := [z^{\uparrow}]_{2^{j+1}} + 2^j$
13 **else**
14 $\overline{z} := [z^{\uparrow}]_{2^{j+1}} - 2^j$
15 **if** $\overline{z} \geq 2^{j-1} - \beta$ or $\overline{z} \leq -2^{j-1} + \beta$ **then** ▷ Only informative relations.
16 **if** $j \geq \log_2 \beta + 1$ **then**
17 **if** $\overline{z} \geq 2^{j-1} - \beta$ **then** ▷ j-independence Transform., Eq.(3).
18 $\widetilde{z} := \overline{z} - 2^{j-1} + \beta$
19 **else**
20 $\widetilde{z} := \overline{z} + 2^{j-1} - \beta$
21 Store $(\mathbf{c}, \widetilde{z})$ in $\widetilde{L}$.
22 Let $m := |\widetilde{L}|$. Create an $(m \times n)$-matrix $\mathbf{C}$ with rows $\mathbf{c} \in [\pm 1]^n_\tau$ from $\widetilde{L}$.
23 Create vector $\mathbf{z}$, whose entries are the $\widetilde{z}$'s from $\widetilde{L}$.
24 Apply ordinary least squares regression to $(\mathbf{C}, \mathbf{z})$, to obtain an estimate $\hat{\mathbf{x}} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z} \in \mathbb{Q}^n$ for the partial key $\mathbf{x}$.
25 Compute the subkey $\mathbf{x} := \lfloor \hat{\mathbf{x}} \rceil \in \mathbb{Z}^n$ of $\mathbf{s}_1$ via coordinate-wise rounding.
26 Recover the full LWE secret $\mathbf{s}_1$ via lattice reduction. ▷ See Section 4.3.
27 **return** $\mathbf{s}_1$.

Putting Things Together: Our Leaky-Signature-LWE Attack. Summarizing the previous sections, we finally obtain our full algorithm, as depicted in Algorithm 1.

5 Analysis of Ordinary Least Squares Regression

In this section, we provide a useful lower bound for the required amount m of informative relations such that Algorithm 1 succeeds in line 25 to recover the

correct subkey $\mathbf{x}$ of $\mathbf{s}_1$. For estimating $\mathbf{x}$, we use ordinary least squares regression (OLS), the most commonly used form of linear regression[4].

Recall that Algorithm 1 computes in line 24 the estimate $\hat{\mathbf{x}} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z}$. The following Lemma 7 states the expectation of $\mathbf{C}^T\mathbf{C}$ and shows that we can actually compute the inverse $(\mathbf{C}^T\mathbf{C})^{-1}$.

Lemma 7. *Let $\mathbf{C}$ be an $(m \times n)$ matrix with rows independently sampled from $[\pm 1]_\tau^n$. For $m > n$, it holds that*

$$\mathbb{E}\left[\mathbf{C}^T\mathbf{C}\right] = \frac{m\tau}{n}\mathbf{I}_n. \tag{17}$$

Each entry of $\mathbf{C}^T\mathbf{C}$ converges to its expected value 0 (off-diagonal) or $\frac{m\tau}{n}$ (on-diagonal) exponentially fast in m. For $m > 2n\ln(n)$, it holds that

$$\Pr[\mathbf{C}^T\mathbf{C} \text{ is invertible }] > 1 - ne^{-\frac{m}{2n}}. \tag{18}$$

Proof. See Appendix B.

The following estimate provides us a lower bound for the amount of required informative relations.

Estimate 8. *Let $(\mathbf{c}, z, y_i)$ with $\mathbf{c} \in [\pm 1]_\tau^n$ be leaky signatures, where the corresponding LWE secret key $\mathbf{s}_1$ has coordinates from $[\pm\eta]$. Let $\beta = \eta \cdot \tau$.*

Then the Leaky-Signature-LWE Attack (Algorithm 1) succeeds on input $(\mathbf{c}, z, y_i)$ to recover the correct subkey $\mathbf{x}$ of $\mathbf{s}_1$ in line 25 with probability at least $1 - \delta$ provided that we have at least

$$m := |\tilde{L}| > \frac{2n}{3\tau}((2\beta+1)^2 - 1)\ln\left(\frac{2n}{\delta}\right) \tag{19}$$

informative relations.

Justification. Algorithm 1 computes in $\tilde{L}$ relations of the form $(\mathbf{C}, \mathbf{z})$ with $\mathbf{C}$ an $(m \times n)$ matrix, where the rows are independent samples of $\mathbf{c} \in [\pm 1]_\tau^n$, and

$$\mathbf{z} = \mathbf{C}^T\mathbf{x} + \mathbf{y}.$$

In Line 24 we compute an OLS estimate

$$\begin{aligned}\hat{\mathbf{x}} &= (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z} \\ &= (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T(\mathbf{C}\mathbf{x} + \mathbf{y}) \\ &= \mathbf{x} + (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}.\end{aligned}$$

According to Theorem 3, the coordinates of $\mathbf{y}$ are i.i.d. samples from a uniform distribution over $[\pm\beta]$. Therefore, the LWE error $\mathbf{y}$ is zero in expectation (i.e. $\mathbb{E}[\mathbf{y}] = 0$), is uncorrelated to $\mathbf{C}$ (i.e. $\mathbb{E}[\mathbf{y}|\mathbf{C}] = 0$), and its covariance matrix

[4] For further information about OLS in general, we refer to the text book [SL03].

is $\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \sigma_y^2\mathbf{I}$ with σ_y^2 the variance of a discrete uniform distribution, i.e., $\sigma_y^2 = \frac{(2\beta+1)^2-1}{12}$.

It is a well established fact [SL03] that the OLS estimate is unbiased under these three properties, which means that $\mathbb{E}[\hat{\mathbf{x}}] = \mathbf{x}$. That is, the computed estimate $\hat{\mathbf{x}}$ is indeed the desired subkey $\mathbf{x}$ in expectation.

Now, we have to show that the estimate $\hat{\mathbf{x}}$ converges with increasing m coordinate-wise to the subkey $\mathbf{x}$.

Using $\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \sigma_y^2\mathbf{I}_m$, the covariance of the estimation error $\hat{\mathbf{x}} - \mathbf{x}$ can therefore be characterized as

$$\begin{aligned}
\mathrm{Cov}[\hat{\mathbf{x}} - \mathbf{x}] &= \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T] \\
&= \mathbb{E}\left[\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\right)\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\right)^T\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{C},\mathbf{y}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\mathbf{y}^T\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\right]. \\
&= \sigma_y^2 \mathop{\mathbb{E}}_{\mathbf{C}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\right] \\
&= \sigma_y^2\,\mathbb{E}[(\mathbf{C}^T\mathbf{C})^{-1}] \\
&\approx \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n. \qquad (20)
\end{aligned}$$

Note, that the last step follows from Lemma 7: Each entry in $\mathbf{C}^T\mathbf{C}$ converges exponentially fast to those of a scaled identity matrix, and as a consequence the same rate of convergence applies to the entries of the inverse, i.e., $(\mathbf{C}^T\mathbf{C})^{-1} \to \frac{n}{m\tau}I_n$. We conclude that independently the variance for each coordinate of $\hat{\mathbf{x}} - \mathbf{x}$ is bounded by approximately $\frac{n\sigma_y^2}{m\tau}$.

We proceed by investigating each coordinate j separately and observe that each component of the estimation error $(\mathbf{x} - \hat{\mathbf{x}})_j$ satisfies

$$(\mathbf{x} - \hat{\mathbf{x}})_j = \sum_{i=1}^{m}((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T)_{ji}\,\mathbf{y}_i,$$

which is a sum of independent random variables $\mathbf{y}_i$ weighted by some entry of $(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T$. We can apply the Lindeberg-Feller central limit theorem [Fel91] which states that the sum of independent random variables with finite variance satisfying the Lindeberg condition converges to a Gaussian distribution.

In our case, the random variables of interest are $Y_i := ((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T)_{ji}\,\mathbf{y}_i$ with $\mathbb{E}[Y_i] = 0$ and variance $\mathrm{Var}[Y_i] = \sigma_i^2 > 0$. With $\tilde{\sigma}_m^2 := \sum_{i=1}^m \sigma_i^2$, the Lindeberg condition reads

$$\forall\epsilon > 0 : \lim_{m\to\infty}\frac{1}{\tilde{\sigma}_m^2}\sum_{i=1}^{m}\mathbb{E}\left[Y_i^2\mathbb{1}_{\{|Y_i|>\epsilon\tilde{\sigma}_m\}}\right] \to 0. \qquad (21)$$

As $|Y_i| = |((\mathbf{C}^T\mathbf{C})^T\mathbf{C}^T)_{ji}\mathbf{y}_i| \in \mathcal{O}(1/m)$ (recall that $(\mathbf{C}^T\mathbf{C})^{-1} \to \frac{n}{m\tau}\mathbf{I}$, which follows from Lemma 7 and entries in $\mathbf{C}^T$ and $\mathbf{y}_i$ are bounded), but $\tilde{\sigma}_m \in \mathcal{O}(1/\sqrt{m})$

(which follows from Equation (20) , where $\tilde{\sigma}_m^2 \approx \frac{n\sigma_y^2}{m\tau}$ is established), we conclude that for any $\epsilon > 0$ there exists $m_0 \in \mathbb{N}$ such that $|Y_i| < \epsilon\tilde{\sigma}_m$, and, thus, the indicator function in Equation (21) will almost always evaluate to 0 for m large enough. Thus, Lindeberg's condition holds and the Lindeberg-Feller version of the central limit theorem applies, i.e., the estimation error converges to a normal distribution with

$$\hat{\mathbf{x}} - \mathbf{x} \sim \mathcal{N}\big(0, \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n\big) \ .$$

This enables us to apply a Gaussian tail bound for each component

$$\Pr[|\hat{\mathbf{x}}_i - \mathbf{x}_i| > \delta_1] \leq 2\exp\left(-\frac{\delta_1^2 m\tau}{2n\sigma_y^2}\right),$$

and utilize the union bound to cover all n dimensions

$$\Pr[\exists i : |\hat{\mathbf{x}}_i - \mathbf{x}_i| > \delta_1] \leq 2n\exp\left(-\frac{\delta_1^2 m\tau}{2n\sigma_y^2}\right).$$

In order to round $\hat{\mathbf{x}}$ successfully to $\mathbf{x}$, we need to establish that no coordinate of $\hat{\mathbf{x}}$ has an error of more than $\delta_1 = \frac{1}{2}$, therefore bounding the ℓ_∞-norm. If we want to succeed with a probability of δ, we can rearrange for the sample complexity:

$$\begin{aligned}\Pr\left[\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_\infty > \frac{1}{2}\right] &\leq 2n\exp\left(-\frac{m\tau}{8n\sigma_y^2}\right) \leq \delta \\ \implies m &\geq \frac{8n\sigma_y^2}{\tau}\ln\left(\frac{2n}{\delta}\right).\end{aligned}$$

Plugging in $\sigma_y^2 = \frac{(2\beta+1)^2-1}{12}$ gives the desired estimate. ◇

Table 3. Average run time for ordinary least squares regression, experimentally required number m of informative relations in Algorithm 1 (compare with Figure 8), our estimate Eq. (19) from Estimate 8, and the estimate Eq. (22) from [LZS+20].

Scheme	regression [seconds]	exp. m [million]	Eq. (19) [million]	Eq. (22) [million]
ML-DSA-44	4	0.50	0.60	11
ML-DSA-87	7	0.75	0.92	17
ML-DSA-65	40	2.4	3.0	54
Ring-LWE-1024	140	2.6	3.4	52

Tightness of Our Bound and Comparison with [LZS+20]. In Table 3, we compare our bound m for the required number of informative relations from Equation (19) in Estimate 8 to our experimental data from Section 6, and to the bound in [LZS+20]. The bound in [LZS+20] was derived from the work in [BDE+18]. For a fair comparison, we instantiate both bounds with success probability $\frac{1}{2}$, which gives for [LZS+20] the formula

$$m \geq \frac{32}{3} \frac{n}{\tau}((2\beta+1)^2 - 1)\ln(2n). \tag{22}$$

From Table 3, we see that our bound from Equation (19) accurately matches the experimentally required amount of informative relations from column exp. m, and that we significantly improve over Equation (22). Our improvement comes from tailoring our analysis in Estimate 8 to leaky LWE signatures, whilst the analysis in [LZS+20] relies on simple sub-Gaussian tail bounds.

Reframing Subkey Recovery as a Lattice Problem. Following the framework of [BDE+18], we applied least squares regression with rounding to solve for the secret subkey $\mathbf{x}$. As pointed out in [BDE+18], the regression approach may also be phrased in lattice language as follows.

Consider the lattice L spanned by the columns of the basis $\mathbf{C}^T\mathbf{C}$. Then, given the target $\mathbf{C}^T\mathbf{z}$, one may solve a close lattice vector problem as

$$\mathbf{C}^T\mathbf{C}\mathbf{x} = \mathbf{C}^T\mathbf{z} - \mathbf{C}^T\mathbf{y}, \tag{23}$$

i.e., the lattice vector $\mathbf{C}^T\mathbf{C}\mathbf{x}$ is $\mathbf{C}^T\mathbf{y}$-close to the target $\mathbf{C}^T\mathbf{z}$. However, as we showed in Lemma 7, L is almost orthogonal, since it asymptotically converges to a scaled unit matrix. Therefore, multiplying $\mathbf{C}^T\mathbf{z}$ by $(\mathbf{C}^T\mathbf{C})^{-1}$ from the left and rounding to the nearest integer vector provides a close vector solution. This approach is known as *Babai's rounding algorithm*, and the approach is in fact identically to our ordinary least squares regression, see lines 24 & 25 in Algorithm 1.

[BDE+18] also shows that asymptotically no other close vector problem algorithm performs better, including *Babai's nearest plane algorithm*. We experimentally tried to reduce the number of required samples m for subkey reconstruction using Babai's nearest plane algorithm instead of rounding. Our experiments indicate that nearest plane does not help to decrease m.

6 Experimental Results

We implemented our Leaky-Signature-LWE attack (Algorithm 1), and ran it on an AMD EPYC 7763 with 1 TB of RAM, as well as on an AMD EPYC 7742 with 2TB of RAM. Each EPYC was equipped with 128 cores.

We attacked a Lyubashevsky ring LWE signature with $n = 1024$ for full secret key recovery, and all three Dilithium parameter sets ML-DSA-44, ML-DSA-65, and ML-DSA-87, each for a 256-dimensional subkey recovery. In our

experiments, we generated sufficiently many leaky signatures $(\mathbf{c}, z, y_j)$ with a single constantly stuck randomness leakage bit $y_j = 0$.

Our results are depicted in Figures 8 to 10.

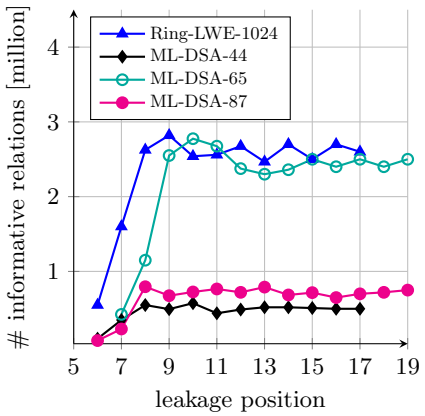

Fig. 8. Required informative relations to recover the secret key in Ring-LWE-1024 or the subkey in Dilithium.

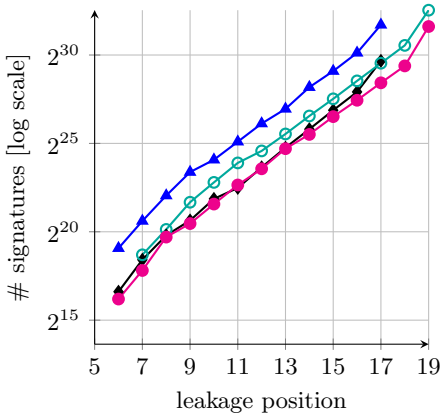

Fig. 9. Required leaky signatures to recover the secret key in Ring-LWE-1024 or the subkey in Dilithium.

Amount of Informative Relations. Figure 8 nicely illustrates the effect of our relation transformation (Section 4.2), which makes our attack independent from the leakage position j. Therefore, the required amount of informative relations for successful subkey recovery essentially remains constant. By Theorem 3, informative relations have an error distributed uniformly in $[\pm\beta]$. A smaller error is easier to correct for linear regression. Thus, the smaller β, the fewer relations should be required.

This observation is in line with the results in Figure 8. ML-DSA-44 with its small $\beta = 78$ (see Table 1) requires the smallest amount of 0.5 million informative relations. For ML-DSA-87 with $\beta = 120$, we require 0.8 million informative relations. For ML-DSA-65 with the largest $\beta = 198$ the number of informative relations increases to 2.4 million.

Although Ring-LWE-1024 has $\beta = 78$, it still requires 2.6 million relations, because linear regression has to recover $n = 1024$ coordinates of the secret key.

Amount of Signatures. Figure 9 illustrates the total amount of signatures that we require for (sub-)key recovery. We observe from Figure 9 that for all four attacked schemes the number of signatures roughly doubles with each increase of the leakage position j by 1. This is expected, since the number of *zero-knowledge* relations, that we sort out in Algorithm 1, also roughly doubles.

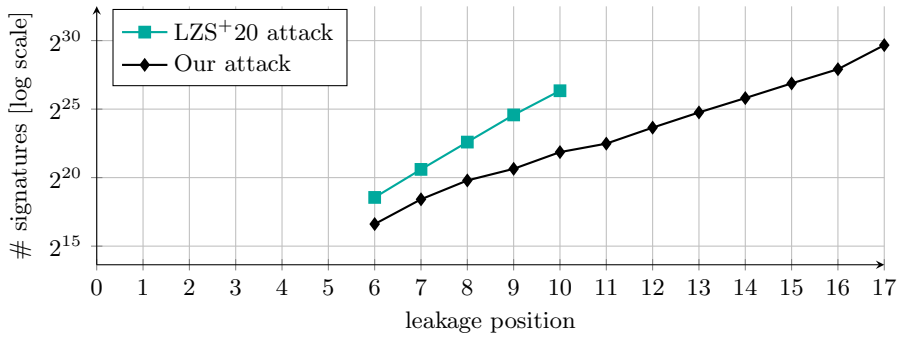

Fig. 10. Required amount of leaky signatures to recover the secret subkey in Dilithium-II (ML-DSA-44).

Interestingly, the amount of signatures that we require is almost alike for ML-DSA-44 and ML-DSA-87, although by Figure 8 we need significantly more informative relations for ML-DSA-87. As can be seen by Figure 4, the larger β, the larger the proportion of *informative relations* among all signatures. This implies that we generate for ML-DSA-87 (with $\beta = 120$) more informative relations than for ML-DSA-44 (with $\beta = 78$) from the same amount of signatures.

The same effect can be observed when comparing the required amount of signatures of Ring-LWE-1024 (with $\beta = 78$) and ML-DSA-65 (with $\beta = 198$). Although both require roughly the same amount of informative relations, we obtain this amount with less ML-DSA-65 signatures.

Comparison with [LZS+20]. Although in absolute numbers we still require for high-order leakage positions j a huge amount of leaky signatures, we would like to stress again that our attack processes all leaky signatures *on the fly*, and stores only informative relations. This is in contrast to the [LZS+20] attack, which produces for every signature a relation that has to be stored for feeding it to linear regression, making [LZS+20]'s attack infeasible for large j.

Figure 10 shows that we do not only improve in run time and memory, but also in the total number of required signatures. While the [LZS+20] attack roughly has a factor 4 increase of required leaky signatures per position, our increase is only by a factor 2, as can be observed by the smaller slope of our attack in Figure 10. The smaller slope is a result of two improvements in our attack. First, we sort out *zero-knowledge* relations, and second we reduce via our novel transformation the size of the error in the relations.

7 Further Reducing the Amount of Signatures

We know from Section 6 that a 256-dimensional subkey $\mathbf{x}$ recovery for ML-DSA-44 requires roughly 500.000 signatures. In this section, we explore what happens

if we do not obtain the required amount of ML-DSA-44 signatures to fully recover the subkey via linear regression. Let us say, we receive only 450.000 or 250.000 signatures. Does that mean that we simply cannot run our Algorithm 1?

We will see in the following that we still obtain useful information that helps us to reduce the complexity of a lattice attack on ML-DSA-44.

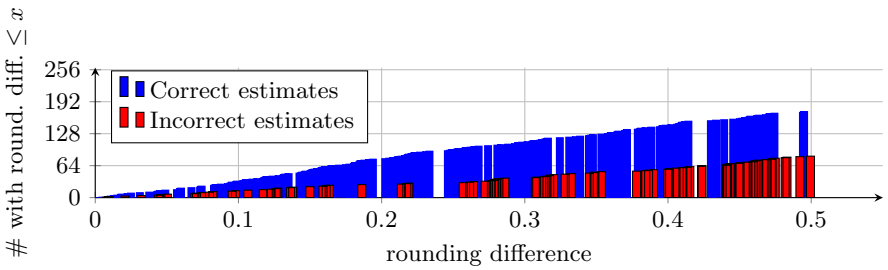

Fig. 11. ML-DSA-44, rounding differences for 50.000 informative relations.

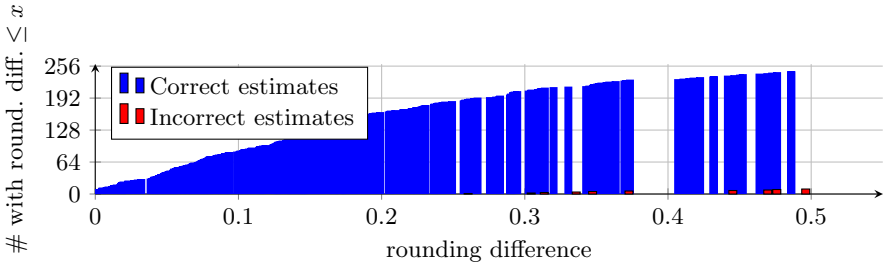

Fig. 12. ML-DSA-44, rounding differences for 250.000 informative relations.

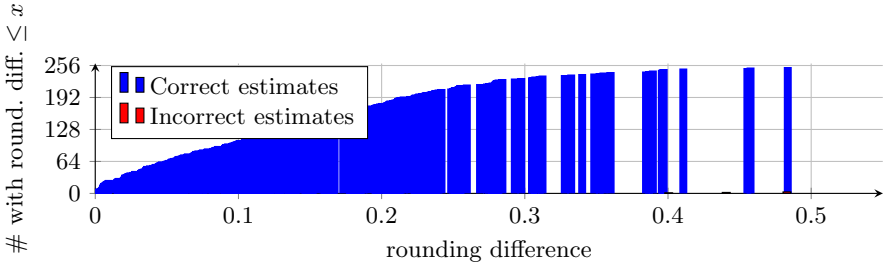

Fig. 13. ML-DSA-44, rounding differences for 450.000 informative relations.

Linear Regression and Rounding. In principle, Algorithm 1 may run linear regression with any available number of relations for ML-DSA-44. Linear

regression will always output an estimate $\hat{\mathbf{x}} \in \mathbb{Q}^{256}$ of the desired subkey. Our Estimate 8 only guarantees that rounding $\hat{\mathbf{x}}$ coordinate-wise to the nearest integer indeed yields the desired subkey $\mathbf{x}$ when we use a sufficient amount of relations. This in turn does not imply that for an insufficient amount of relations the estimate $\hat{\mathbf{x}} \in \mathbb{Q}^{256}$ is completely off.

We ran Algorithm 1 on ML-DSA-44 with 50.000, 250.000, and 450.000 informative relations, and depicted the results in Figures 11 to 13, respectively.

Let us have a closer look at Figure 11 with only 50.000 relations. We see that after rounding already 170 out of the 256 coefficients of $\mathbf{x}$ are correctly determined by linear regression. If we use 250.000 relations, then we obtain 240 correctly determined coefficients, and for 450.000 relations we even obtain 250 correct coordinates of $\mathbf{x}$.

Thus, linear regression has the remarkable property that it correctly identifies already a large fraction of all coordinates of the secret $\mathbf{x}$ with significantly less relations than required to fully determine $\mathbf{x}$. In other words, the number of correctly identified coordinates quite quickly convergences to n.

Scoring Coordinates. While the property that linear regression identifies a large fraction of coordinates correctly looks promising, it is unclear how to exploit this algorithmically. Enumerating the incorrect coordinates still seems infeasible, since we do not know the positions of the errors.

Instead, we propose the following *scoring* of coordinates. For every coordinate in the estimate $\hat{\mathbf{x}} \in \mathbb{Q}^n$ we determine its remainder $r_i = \hat{\mathbf{x}}_i - \lfloor \hat{\mathbf{x}}_i \rceil$, i.e., r_i is the difference to the next integer. Intuitively, the smaller the score r_i, the more likely is the rounded value $\lfloor \hat{\mathbf{x}}_i \rceil$ indeed the correct coordinate $\mathbf{x}_i$.

We see in Figure 12 that such a scoring is a useful measure. Among the incorrectly identified coordinates a remainder of $r_i = 0.26$ is the minimal score. So we can identify all coordinates with score smaller than the threshold $r_i = 0.26$ as correct. These would still be an amount of 192 coordinates from $\mathbf{x}$.

Now, we may apply the *LWE with side information framework* from Section 4.3. From Figure 5 we see that by using 192 known coordinates of the secret key, the lattice complexity of ML-DSA-44 drops to below 100 bit.

Acknowledgments. We thank Phong Nguyen for helpful discussions.

Nicolai Kraus is funded by the German Federal Ministry of Education and Research (BMBF) project PQ-CCA. Julian Nowakowski is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) grant 465120249. Alexander May is supported by DFG under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

Disclosure of Interests. The authors have no competing interests.

References

ABC+22. Martin R Albrecht, Daniel J Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo Von Maurich, Rafael Misoczki, Ruben Niederhagen, et al. Classic mceliece: conservative code-based cryptography. 2022.

AH21. Martin R. Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. In Anne Canteaut and Francois-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 528–558. Springer, Cham, October 2021.

Aka09. Adi Akavia. Solving hidden number problem with one bit oracle and advice. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 337–354. Springer, Berlin, Heidelberg, August 2009.

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

BDE+18. 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 2018, Part I*, volume 11272 of *LNCS*, pages 494–524. Springer, Cham, December 2018.

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

CFS01. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174. Springer, Berlin, Heidelberg, December 2001.

CNE+14. Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. On the practical exploitability of dual EC in TLS implementations. In Kevin Fu and Jaeyeon Jung, editors, *USENIX Security 2014*, pages 319–335. USENIX Association, August 2014.

CVE. Cve-2024-31497. https://nvd.nist.gov/vuln/detail/CVE-2024-31497. Accessed: 30.04.2024.

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

DFPS23. Julien Devevey, Pouria Fallahpour, Alain Passelègue, and Damien Stehlé. A detailed analysis of Fiat-Shamir with aborts. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 327–357. Springer, Cham, August 2023.

DGHK23. Dana Dachman-Soled, Huijing Gong, Tom Hanson, and Hunter Kippen. Revisiting security estimation for LWE with hints from a geometric perspective. In Helena Handschuh and Anna Lysyanskaya, editors,

CRYPTO 2023, Part V, volume 14085 of *LNCS*, pages 748–781. Springer, Cham, August 2023.

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

EFGT17. Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1857–1874. ACM Press, October / November 2017.

Fel91. William Feller. *An introduction to probability theory and its applications, Volume 2*, volume 81. John Wiley & Sons, 1991.

FGUO+13. Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate mceliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

FS09. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, Berlin, Heidelberg, December 2009.

Gal13. Steven D Galbraith. Space-efficient variants of cryptosystems based on learning with errors. *url: https://www. math. auckland. ac. nz/˜ sgal018/compact-LWE. pdf*, 2013.

GGH97. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131. Springer, Berlin, Heidelberg, August 1997.

GJSS01. Craig Gentry, Jakob Jonsson, Jacques Stern, and Michael Szydlo. Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, December 2001.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

GS02. Craig Gentry and Michael Szydlo. Cryptanalysis of the revised NTRU signature scheme. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 299–320. Springer, Berlin, Heidelberg, April / May 2002.

HGS01. Nick A Howgrave-Graham and Nigel P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, 23:283–290, 2001.

HHP+03. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140. Springer, Berlin, Heidelberg, April 2003.

HM17. Gottfried Herold and Alexander May. LP solutions of vectorial integer subset sums — cryptanalysis of Galbraith's binary matrix LWE. In Serge Fehr,

editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 3–15. Springer, Berlin, Heidelberg, March 2017.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, June 1998.

HPS01. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NSS: An NTRU lattice-based signature scheme. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 211–228. Springer, Berlin, Heidelberg, May 2001.

HR07. Martin Hlavác and Tomás Rosa. Extended hidden number problem and its cryptanalytic applications. In Eli Biham and Amr M. Youssef, editors, *SAC 2006*, volume 4356 of *LNCS*, pages 114–133. Springer, Berlin, Heidelberg, August 2007.

HR23. Nadia Heninger and Keegan Ryan. The hidden number problem with small unknown multipliers: Cryptanalyzing MEGA in six queries and other applications. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 147–176. Springer, Cham, May 2023.

HRSS17. Andreas Hülsing, Joost Rijneveld, John M Schanck, and Peter Schwabe. Ntru-hrss-kem-submission to the nist post-quantum cryptography project. 2017.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009.

Lyu24. Vadim Lyubashevsky. Basic lattice cryptography: The concepts behind kyber (ML-KEM) and dilithium (ML-DSA). Cryptology ePrint Archive, Report 2024/1287, 2024.

LZS+20. Yuejun Liu, Yongbin Zhou, Shuo Sun, Tianyu Wang, Rui Zhang, and Jingdian Ming. On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage. *IEEE Transactions on Information Forensics and Security*, 16:1868–1879, 2020.

MBA+21. Robert Merget, Marcus Brinkmann, Nimrod Aviram, Juraj Somorovsky, Johannes Mittmann, and Jörg Schwenk. Raccoon attack: Finding and exploiting most-significant-bit-oracles in TLS-DH(E). In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 213–230. USENIX Association, August 2021.

McE78. Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

MN23. Alexander May and Julian Nowakowski. Too many hints - when LLL breaks LWE. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part IV*, volume 14441 of *LNCS*, pages 106–137. Springer, Singapore, December 2023.

NR09. Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *Journal of Cryptology*, 22(2):139–160, April 2009.

NS02. Nguyen and Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptology*, 15:151–176, 2002.

NS03. Phong Q Nguyen and Igor E Shparlinski. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Designs, codes and cryptography*, 30:201–217, 2003.

OVCG24. Paco Azevedo Oliveira, Andersson Calle Viera, Benoît Cogliati, and Louis Goubin. Uncompressing dilithium's public key. *IACR Cryptol. ePrint Arch.*, page 1373, 2024.

QLZ+22. Zehua Qiao, Yuejun Liu, Yongbin Zhou, Jingdian Ming, Chengbin Jin, and Huizhong Li. Practical public template attacks on crystals-dilithium with randomness leakages. *IEEE Transactions on Information Forensics and Security*, 18:1–14, 2022.

Rya18. Keegan Ryan. Return of the hidden number problem. *IACR TCHES*, 2019(1):146–168, 2018.

Sch87. Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2-3):201–224, 1987.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

SL03. George A. F. Seber and Alan J. Lee. *Linear regression analysis*. Wiley series in probability and statistics. Wiley-Interscience, Hoboken, NJ, 2. ed. edition, 2003.

Tro12. Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012.

XSWH22. Jun Xu, Santanu Sarkar, Huaxiong Wang, and Lei Hu. Improving bounds on elliptic curve hidden number problem for ECDH key exchange. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 771–799. Springer, Cham, December 2022.

A Probabilistic Leakage

Throughout this paper, we assume the attacker has access to a randomness bit y_j for each signature. In practice, this bit may be affected by side-channel noise, such that it is correct with probability p and incorrect with probability $1 - p$.

This probabilistic leakage model was considered in [LZS+20] in the context of their attack. The authors observed that when the leak bits are always incorrect (i.e., the attacker obtains $\bar{y}_j \coloneqq 1 \oplus y_j$), their attack recovers the negated secret $-\mathbf{x}$. Further, for leakage probabilities $p \in (0.5, 1]$, the secret $\mathbf{x}$ can be recovered by scaling the estimate $\hat{\mathbf{x}}$ by $1/(2p - 1)$.

We observed experimentally that the same applies to our attack. However, we do not yet have a formal proof of this effect and leave it as an open problem. See Algorithm 2 for the adapted algorithm, which reduces to Algorithm 1 when initialized with leakage probability $p = 1$.

In our experiments, we successfully recovered the secret subkey $\mathbf{x}$ for leakage probabilities $p \geq 0.55$.

Algorithm 2: Leaky-Signature-LWE Attack for Probabilistic Leakage

Input : List L relations $(\mathbf{c}, z, y_j)$ with
challenge vector $\mathbf{c} \in [\pm 1]_\tau^n$,
signature coefficient $z \in [\pm(\gamma - \beta)]$,
knowledge of a bit of the randomness $y_j \in [0, 1]$,
leakage index j,
leakage probability $p \in (0.5, 1]$,
public key $(\mathbf{A}, \mathbf{t})$.

Output: LWE secret $\mathbf{s}_1 \in ([\pm\eta]^n)^\ell$.

Initialize empty list $\widetilde{L}$.
for $(\mathbf{c}, z, y_j) \in L$ **do**
  if $[z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$ **then** ▷ Normal form computation, Eq.(6)&(7).
    $z^{\uparrow} := z + 2^{j-1}$
    $b_j := y_j$
  else
    $z^{\uparrow} := z - 2^{j-1}$
    $b_j := y_j \oplus 1$
  if $b_j = 1$ **then** ▷ Relation extraction $\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y^{\uparrow}]_{2^j}$, Eq.(10).
    $\overline{z} := [z^{\uparrow}]_{2^{j+1}}$
  else if $[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$ **then**
    $\overline{z} := [z^{\uparrow}]_{2^{j+1}} + 2^j$
  else
    $\overline{z} := [z^{\uparrow}]_{2^{j+1}} - 2^j$
  if $2^{j-1} + \beta > \overline{z} \geq 2^{j-1} - \beta$ or $-2^{j-1} - \beta < \overline{z} \leq -2^{j-1} + \beta$ **then**
    if $j \geq \log_2 \beta + 1$ **then**
      if $\overline{z} \geq 2^{j-1} - \beta$ **then** ▷ j-independence Transform., Eq.(3).
        $\widetilde{z} := \overline{z} - 2^{j-1} + \beta$
      else
        $\widetilde{z} := \overline{z} + 2^{j-1} - \beta$
      Store $(\mathbf{c}, \widetilde{z})$ in $\widetilde{L}$.
Let $m := |\widetilde{L}|$. Create an $(m \times n)$-matrix $\mathbf{C}$ with rows $\mathbf{c} \in [\pm 1]_\tau^n$ from $\widetilde{L}$.
Create vector $\mathbf{z}$, whose entries are the $\widetilde{z}$'s from $\widetilde{L}$.
Apply ordinary least squares regression to $(\mathbf{C}, \mathbf{z})$, to obtain an estimate $\hat{\mathbf{x}} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z} \in \mathbb{Q}^n$ for the partial key $\mathbf{x}$.
Compute the subkey $\mathbf{x} := \lfloor \frac{1}{2p-1}\hat{\mathbf{x}} \rceil \in \mathbb{Z}^n$ of $\mathbf{s}_1$ via coordinate-wise rounding.
Recover the full LWE secret $\mathbf{s}_1$ via lattice reduction. ▷ See Section 4.3.
return $\mathbf{s}_1$.

B Proof of Lemma 7

Proof. We first delve into some properties of $\mathbf{C}^T\mathbf{C} \in \mathbb{R}^{m\times n}$ which will be useful for this proof:

- For each column $\mathbf{C}_{:i}$ it holds, that $\mathbb{E}[\mathbf{C}_{:i}^T\mathbf{C}_{:i}] = \mathbb{E}[\sum_{k=1}^m \mathbf{C}_{ki}^2] = m \cdot 1 \cdot \frac{\tau}{n}$ since we can assume, that each element in one column is i.i.d. and with probability $\frac{\tau}{n}$ to be non-zero.
- For all columns $i \neq j$ of $\mathbf{C}$, i.e. $\mathbf{C}_{:i}$ and $\mathbf{C}_{:j}$, we have $\mathbb{E}[\mathbf{C}_{:i}^T\mathbf{C}_{:j}] = \mathbb{E}[\sum_{k=1}^m \mathbf{C}_{ki}\mathbf{C}_{kj}] = m(1 \cdot 2(\frac{\tau}{2n})^2 - 1 \cdot 2(\frac{\tau}{2n})^2 + 0) = 0$. Entries are from $\{-1, 0, 1\}$ where both non-zero elements occur with the same probability.

We first note using the above insights, that

$$\mathbb{E}\left[\mathbf{C}^T\mathbf{C}\right] = \begin{pmatrix} \mathbb{E}[\mathbf{C}_{:1}^T\mathbf{C}_{:1}] & \mathbb{E}[\mathbf{C}_{:1}^T\mathbf{C}_{:2}] & \mathbb{E}[\mathbf{C}_{:1}^T\mathbf{C}_{:n}] \\ \vdots & \mathbb{E}[\mathbf{C}_{:i}^T\mathbf{C}_{:i}] & \vdots \\ \vdots & \cdots & \mathbb{E}[\mathbf{C}_{:n}^T\mathbf{C}_{:n}] \end{pmatrix} = \frac{m\tau}{n}\mathbf{I}_n. \tag{24}$$

This shows Equation (17).

Using Hoeffding's inequality for some $\delta_2 > 0$, we show that each diagonal element converges to $\frac{\tau}{n}$ exponentially fast in m as

$$\Pr\left[\frac{1}{m}\left|\mathbf{C}_{:i}^T\mathbf{C}_{:i} - m\frac{\tau}{n}\right| \geq \delta_2\right] = \Pr\left[\frac{1}{m}\left|\sum_{k=1}^m \mathbf{C}_{ki}^2 - \mathbb{E}[\mathbf{C}_{ki}^2]\right| \geq \delta_2\right] \leq 2\exp\left(-2m\delta_2^2\right),$$

while each off diagonal element converges to 0 exponentially fast in m),

$$\Pr\left[\frac{1}{m}\left|\mathbf{C}_{:i}^T\mathbf{C}_{:j}\right| \geq \delta_2\right] = \Pr\left[\frac{1}{m}\left|\sum_{k=1}^m \mathbf{C}_{ki}^T\mathbf{C}_{kj} - \mathbb{E}[\mathbf{C}_{:i}^T\mathbf{C}_{:j}]\right| \geq \delta_2\right] \leq 2\exp\left(-\frac{m\delta_2^2}{2}\right).$$

We show that $\mathbf{C}^T\mathbf{C}$ is invertible for sufficiently large m. In order to do this, we use the fact that a matrix is invertible if it only has positive eigenvalues. Therefore, we bound the smallest eigenvalue of $\mathbf{C}^T\mathbf{C}$ away from zero. We first make use of the fact that this matrix can equally be expressed as the sum of outer products of rows $\mathbf{C}_i$, i.e., $\mathbf{C}^T\mathbf{C} = \sum_{i=1}^m \mathbf{C}_i\mathbf{C}_i^T$. From this we yield

$$\mathbf{C}^T\mathbf{C} = \sum_{i=1}^m \underbrace{\left(\mathbf{C}_i\mathbf{C}_i^T\right)}_{\mathbf{S}_i}.$$

The matrices $\mathbf{S}_i$ are symmetric and also independent because rows $\mathbf{C}_i$ are i.i.d., thus we can apply a Matrix Chernoff bound [Tro12, Remark 5.3]. For this, we investigate the spectrum of $\mathbf{S}_i$. Since $\mathbf{S}_i$ is a rank one matrix, $n-1$ eigenvalues are 0, and the remaining eigenvalue that can be bounded using the trace $\lambda(\mathbf{S}_i) = \mathrm{Tr}[\mathbf{S}_i] = \sum_{j=1}^n C_{ij}^2 = \tau$, i.e.,

$$\lambda_{\min}(\mathbf{S}_i) = 0 \qquad \text{and} \qquad \lambda_{\max}(\mathbf{S}_i) \leq \tau \ (:= R).$$

We already know from Equation (24) that

$$\begin{aligned}\mu_{\min} &:= \lambda_{\min}\left(\sum_{i=1}^{m}\mathbb{E}\,\mathbf{S}_i\right) = \lambda_{\min}\left(\sum_{i=1}^{m}\mathbb{E}\left[\mathbf{C}_i\mathbf{C}_i^T\right]\right)\\ &= \lambda_{\min}\left(\mathbb{E}\left[\mathbf{C}^T\mathbf{C}\right]\right) = \lambda_{\min}\left(\frac{m\tau}{n}\mathbf{I}_n\right) = \frac{m\tau}{n}.\end{aligned}$$

Then the Matrix Chernoff bound states that $\forall\delta_3 \in [0,1]$

$$\begin{aligned}&\Pr\left[\lambda_{\min}\left(\sum_{i=1}^{m}\mathbf{S}_i\right) \leq (1-\delta_3)\mu_{\min}\right] \leq n\cdot\exp\big(-\frac{{\delta_3}^2\mu_{\min}}{2R}\big)\\ \implies &\Pr\left[\lambda_{\min}\left(\mathbf{C}^T\mathbf{C}\right) \leq (1-\delta_3)\frac{m\tau}{n}\right] \leq n\cdot\exp\big(-\frac{{\delta_3}^2 m}{2n}\big).\end{aligned}$$

We now set $\delta_3 = 1$ and obtain

$$\Pr\left[\lambda_{\min}\left(\mathbf{C}^T\mathbf{C}\right) \leq 0\right] \leq n\cdot\exp\big(-\frac{m}{2n}\big) \quad \leq 1.$$

Rearranging for m gives Equation (18). This concludes the proof.

□