J Cryptol (2026) 39:28
https://doi.org/10.1007/s00145-026-09584-7

Journal of CRYPTOLOGY

Research Article

One (Noisy) Bit to Rule Them All: Key Recovery from Randomness Leakage in ML-DSA

Simon Damm · Nicolai Kraus · Alexander May
Ruhr-University Bochum, Bochum, Germany
simon.damm@rub.de
nicolai.kraus@rub.de
alex.may@rub.de

Julian Nowakowski
Centrum Wiskunde en Informatica, Amsterdam, The Netherlands
julian.nowakowski@cwi.nl

Jonas Thietke
Ruhr-University Bochum, Bochum, Germany
jonas.thietke@rub.de

Communicated by Damien Stehlé.

Received 30 January 2026 / Revised 13 July 2026 / Accepted 14 July 2026

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 EC-DSA signatures and the new post-quantum standard ML-DSA (aka Dilithium). In practice, EC-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 ML-DSA. We build on, improve and generalize the pioneering leakage attack on ML-DSA by Liu, Zhou, Sun, Wang, Zhang, and Ming. Using a transformation to Integer LWE (ILWE), their attack can recover a 256-dimensional subkey of ML-DSA-44 from leakage in a single bit of $\mathbf{y}$ per signature, in any bit position $j \geq 6$. However, the number of required signatures grows exponentially as 4^j. In this work, we show that not all leaky signatures carry information about the secret subkey. We introduce the notion of *informative* signature relations. This notion allows us to define a preprocessing step, called *filter-and-shift* that leads to ILWE instances that require a smaller sample amount. Unlike the standard ILWE transformation, filter-and-shift exploits the smallness of secret keys, and therefore might be of independent cryptanalytic interest. In comparison to Liu et al., for $j = 6$ we require only a quarter of the signatures and reduce the exponential growth to 2^j. In addition, we show that the secret subkey can be recovered even with a leak bit corrupted by a large amount of noise, in theory up to the maximum of 50%. Experimentally, we still recover the secret with 43% noise, where we need 170 times as many signatures as in

Published online: 10 August 2026

the noise-free setting. 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 recovers the whole key.
Keywords. ML-DSA, Dilithium, Randomness leakage, Key recovery, Side-Channel.

1. Introduction

EC-DSA attacks as a warning. EC-DSA signatures are one of the most important cornerstones for securing authenticity in our digital society. The origin of EC-DSA lies in Schnorr's identification protocol [48] 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. 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 [25,42,43] that the Boneh-Venkatesan lattice-based algorithm for the *hidden number problem* [8] can be utilized to tackle EC-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 [2–4,16,31,51].

A series of devastating real-world attacks [11,30,31,37,46] 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 [10] 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 [38] and lattices [21,28] basically resisted cryptanalytic efforts and just underwent some modernizations [1,32], the story of their signature counterparts [9,26,29] has seen a series of breaks and improvements.

For codes, the well-known McEliece-type CFS signature [9] suffered from slow signing, initial parameters were broken by a Generalized Birthday attack due to Bleichenbacher (see [19]), and later a key distinguishing attack was found [18].

For lattices, the NTRU NSS scheme [29] and its successor NTRUSign [26] have faced effective cryptanalytic attacks [22,24]. Nguyen and Regev [41] showed that the inherently leaked information of GGH and NTRU signatures [21,26] can be exploited to recover the secret key via gradient descent on a multivariate optimization problem. The result of [41] 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 [23] and Lyubashevsky [34] demonstrated how to build lattice-based signatures, provably without secret key leakage. While [23] utilizes the hash-and-sign paradigm, Lyubashevsky [34] 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 [48]. The novel post-quantum standard ML-DSA [33,40] 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 [14,34] 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 [48].

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. In particular, the computation of $\mathbf{z}$ does not involve a modulo-q reduction. 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 ML-DSA-44 parameters (aka Dilithium-II) 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 [27] that *Integer LWE* (ILWE) – i.e., an LWE instance without modular reduction, such as Galbraith's binary matrix LWE [20] – leads to efficient cryptanalysis using methods from linear optimization. In [17], Espitau, Fouque, Gérard and Tibouchi described a side-channel attack on BLISS signatures [13] that via leakage also leads to ILWE equations. Afterwards, the ILWE problem was

systematically studied by Bootle, Delaplace, Espitau, Fouque and Tibouchi [5], 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 [5] 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 $\mathbf{s}$.

By construction, an ML-DSA 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 [34] those equations perfectly protect $\mathbf{s}$, resulting in an *unsolvable* ILWE instance.

Liu-Zhou-Sun-Wang-Zhang-Ming attack [36]. The unsolvable ILWE instance $\mathbf{z} = c\mathbf{s} + \mathbf{y}$ in ML-DSA becomes solvable 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 [36]. For ML-DSA-44, using knowledge of a single leaked bit $y_{i,j}$ in position $j \geq 6$, the work of [36] shows how to transform into a solvable ILWE instance, whose solution is a 256-dimensional subkey $\bar{\mathbf{s}}$ of ML-DSA-44's 1024-dimensional secret $\mathbf{s}$. The authors of [36] apply the ILWE framework of [5] 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 min for recovery of $\bar{\mathbf{s}}$.

However, for bit positions $j > 6$, the attack is significantly less efficient. The reason is that, in the [36] 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 [36] 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.

For $j = 6$, the work of Qiao, Liu, Zhou, Ming, Jin and Li [45] demonstrated the real-world relevance of the [36] attack, by realizing randomness bit leakage via a Public Template Attack on a masked ML-DSA implementation.

1.1. *Our Results*

Our starting point is the [36] attack within the ILWE framework of [5]. However, we strongly deviate from the original description of [36].

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 ML-DSA-44 as an example, [36] 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 slows down linear regression, since the zero-knowledge relations dilute the input data, thereby resulting in a larger amount of required signatures. If we instead feed only informative relations as input to linear regression, then the amount of required signatures already drops significantly.

In the original attack of [36], 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, making the attack more efficient for all bit positions $j \geq 6$. 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.

Easier ILWE via filter-and-shift. Besides *filtering* out the zero-knowledge relations, we further propose to apply a novel *shift* transformation to the informative relations. We prove that this strategy, which we call *filter-and-shift*, allows to solve ILWE using significantly fewer samples. Remarkably, in the standard approach for solving an instance $\mathbf{t} = \mathbf{As} + \mathbf{e}$ [5], the required amount of samples depends only on the distribution of $\mathbf{A}$ and $\mathbf{e}$. In contrast, ours also takes the distribution of $\mathbf{s}$ into account, thereby significantly reducing the number of required samples. We therefore believe that our filter-and-shift approach might be of independent interest for other cryptanalytic applications.

Subkey vs full secret key recovery. 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 ML-DSA-44 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 [12].

Noisy leakage. In practical side-channel attacks, a leak bit is often corrupted by noise, i.e., the leak bit gets flipped with some error rate $p \in [0, 0.5)$. We analyze this noisy leakage setting and show that the attack remains effective. Our attack uses knowledge of p, but, building on [7], we show that even an unknown p can be effectively estimated by an attacker.

In theory, a secret subkey can still be recovered for any error rate below the theoretical maximum of 50%. In experiments, we successfully recover the secret subkey for both known and unknown error rates p of up to 43%. Notably, recovery from an unknown, estimated error rate p requires roughly the same number of samples as for known p.

Analysis of required relations. The work of [5] describes a very general framework for solving ILWE via linear regression. By an application of the [5] framework, [36] achieve a lower bound for the number of required linear relations to successfully recover a secret

subkey of $\mathbf{s}$ 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 in both the noise-free and in the noisy case.

Organization of the paper. In Sect. 2, we recall Lyubashevsky's identification protocol, that together with the Fiat-Shamir transform results in Lyubashevsky's signature scheme. We define our attack model with leaky LWE signatures in Sect. 3. We solve the problem of key recovery from these leaky signatures by (1) extracting ILWE relations on the secret key, (2) applying filter-and-shift to obtain an easier ILWE instance, and (3) solving the resulting ILWE instance using least squares regression.

Since the filter-and-shift technique is not limited to our attack, we bring it forward to Sect. 4, and postpone the rather technical relation extraction to Sect. 5. For ease of exposition, we first consider only the noise-free case $p = 0$, and afterwards generalize our analysis to arbitrary noise $p \in [0, 0.5)$ in Sect. 6. In Sect. 7, we analyze the correctness and run time of our attack. First, we show that least squares converges to the true secret key, then we derive an estimate of the sample complexity. We provide experimental results in Sect. 8. Lastly, we consider practical aspects of our attack in Sect. 9.

2. Lyubashevsky ID Protocol, Fiat-Shamir with Aborts

In this section, we recall Lyubashevsky's identification protocol [34,35] (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 $[r]_q := r \bmod q$, 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.

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 $\beta := \eta \cdot \tau$.

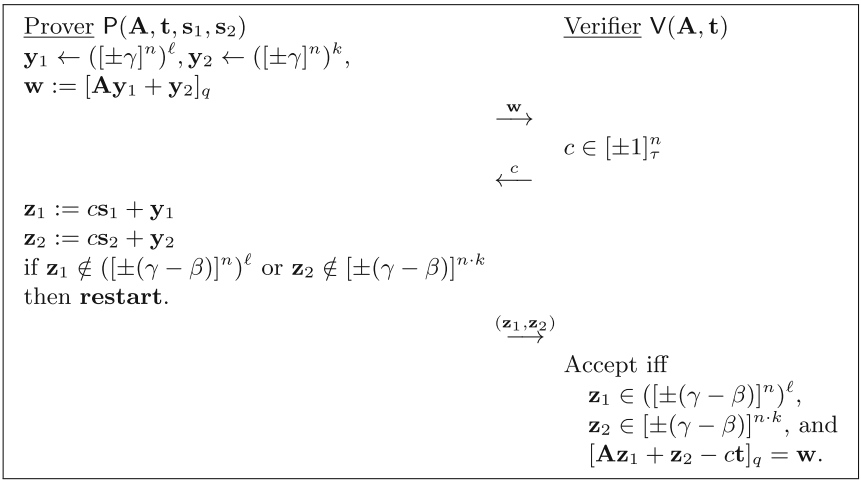

Fig. 1. Lyubashevsky's ID protocol.

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 survey [35, Figure 5], we formally describe the ID Protocol in Fig. 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 computation of $(\mathbf{z}_1, \mathbf{z}_2)$ is not modulo q but over the integers. 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 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.

Differences with ML-DSA. For efficiency purposes, ML-DSA 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

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

reader to Lyubashevsky's survey [35, 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 Fig. 1. We stress, however, that all our results apply to the optimized variant used in ML-DSA 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 [44].

3. Attack Model

We now formally describe our attack model. Consider a Lyubashevsky-type signature scheme based on the ID protocol from Fig. 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 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.

Two's complement. For a word width w, the two's complement stores signed integers x with $-2^{w-1} \le x < 2^{w-1}$ 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 $-2^{w-1} \le [a]_{2^w} < 2^{w-1}$. See Table 2 for an example of the two's complement representations with word width $w = 3$. For a given j, $0 \le j < w$, we call x_j the *j-th bit in the two's complement representation of x*.

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

Table 2. 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

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 $(\mathbf{s}_1, \mathbf{s}_2) \in R^\ell \times R^k$. Write $\mathbf{s}_1 = (s_{1,1}, s_{1,2}, \ldots, s_{1,\ell})$ with $s_{1,i} \in R$. Since each $(R \times R)$-multiplication $c \cdot s_{1,i}$ in $c\mathbf{s}_1 = (c \cdot s_{1,1}, \ldots, c \cdot s_{1,\ell})$ can be seen a matrix–vector multiplication in $\mathbb{Z}^{n\times n} \times \mathbb{Z}^n$, each of the $n \cdot \ell$ coefficients of $\mathbf{z}_1$ yields one linear relation in the LWE secret $\mathbf{s}_1$. Importantly, 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 $(\mathbf{c}, z, y_j) \in [\pm 1]_\tau^n \times \mathbb{Z} \times \{0, 1\}$, that are sampled from the following distribution:

1. Sample $y \leftarrow [\pm\gamma]$, and $\mathbf{c} \leftarrow [\pm 1]_\tau^n$.
2. Set $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$.
3. If $|z| \leq [\pm(\gamma - \beta)]$, output $(\mathbf{c}, z, y_j)$, where y_j is the j-th bit in the binary two's complement representation of y. Otherwise go back to Step 1.

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

3.1. *Recovering the Secret Key* $\mathbf{s}_1$ *from a Subkey* $\mathbf{x}$

The tuples $(\mathbf{c}, z, y_j)$ in Definition 1 only reveal information about the subkey $\mathbf{x}$. To recover the whole key $\mathbf{s}_1$ from $\mathbf{x}$, we follow the *LWE with side information* framework of [12,15,39]. 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 [12] 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 [47] 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 [12] 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 Fig. 2.

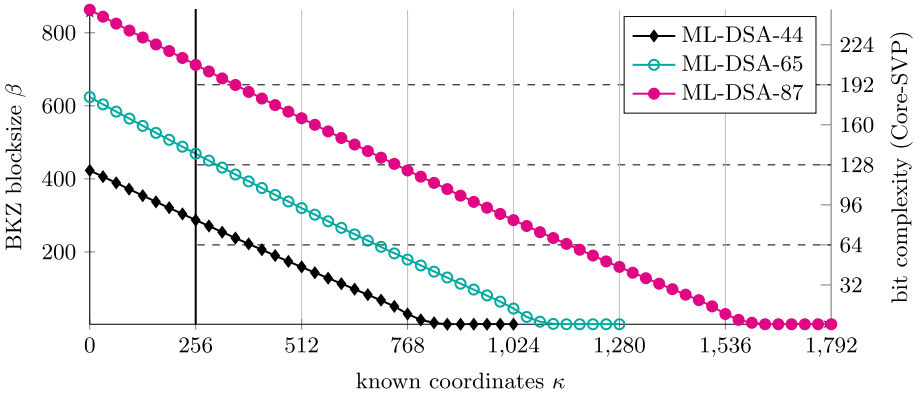

Fig. 2. Required BKZ blocksize to break Dilithium with known coordinates.

For 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, 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.

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. ILWE via Filter-and-Shift Preprocessing

The core of both the randomness leakage attack in [36] and ours consists in solving an ILWE instance $\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y}$ of the following shape:

$$\begin{aligned} &\text{the rows of } \mathbf{C} \in \mathbb{Z}^{m\times n} \text{ are sparse ternary vectors of } \ell_1\text{-norm } \tau,\\ &\text{the entries of } \mathbf{x} \text{ follow the uniform distribution over } [\pm\eta], \text{ and}\\ &\text{the entries of } \mathbf{y} \text{ follow the uniform distribution over } [\pm\gamma_j], \end{aligned} \tag{1}$$

where τ, η and γ_j are some parameters. The parameters τ and η are the same as in the ID protocol from Fig. 1, but γ_j is strictly smaller than γ in Fig. 1. We instead have $\gamma_j \approx 2^j$, where j is the bit leak index from Definition 1.

In Sect. 5, we explain in detail how we extract an ILWE instance $\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y}$ of this shape from leaky Lyubashevsky signatures. However, for now, we treat the explanation of the *extraction* in a black box fashion, and instead first show how to *solve* the resulting ILWE instance. To this end, we introduce a novel algorithm that recovers $\mathbf{x}$ from $(\mathbf{C}, \mathbf{z})$ more efficiently than the standard approach of [5]. Specifically, instead of *directly* applying least squares regression to $(\mathbf{C}, \mathbf{z})$, we propose to first perform a novel preprocessing step. Since ILWE instances as in Equation (1) might also arise in different contexts, we believe that our algorithm for solving them might be of independent interest. We therefore chose to not obscure the algorithm's description with the extraction that is specific to the setting of leaky Lyubashevksy signatures.

Let $\sigma_j^2 := \gamma_j(\gamma_j + 1)/3$ denote the variance of the entries of $\mathbf{y}$, and let us model the distribution of the entries of $\mathbf{C}$ by a zero-mean distribution with variance $\sigma_c^2 := \tau/n$. Then the smallest m for which the standard approach successfully solves such an ILWE instance grows as

$$m = \Theta\left(\frac{\sigma_j^2}{\sigma_c^2}\log(n)\right) = \Theta\left(\frac{\gamma_j^2 n}{\tau}\log(n)\right), \tag{2}$$

see [5, Theorem 4.5]. In contrast, the smallest m for which our novel approach suceeds is

$$m = \Theta\left(\gamma_j \eta n \log(n)\right). \tag{3}$$

We therefore improve over the standard approach whenever

$$\eta < \frac{\gamma_j}{\tau}. \tag{4}$$

Equation (4) can be interpreted as follows. Let us denote the rows of $\mathbf{C}$ by $\mathbf{c}_i$ and set $\beta := \eta\tau$. Then for any $\mathbf{c}_i \in [\pm 1]_\tau^n$, we have

$$|\langle \mathbf{c}_i, \mathbf{x}\rangle| \leq \|\mathbf{c}_i\|_1 \cdot \|\mathbf{x}\|_\infty = \beta.$$

Since Equation (4) is equivalent to $\beta < \gamma_j$, it follows that our approach is superior whenever the inner products $\langle \mathbf{c}_i, \mathbf{x}\rangle$ are small compared to $\gamma_j = \|\mathbf{y}\|_\infty$. In particular, in contrast to the standard approach, ours incorporates the size of the entries of $\mathbf{x}$.

4.1. *Zero-Knowledge Samples*

The ILWE instance $\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y}$ from Equation (1) is closely related to Lyubashevsky's ID protocol from Fig. 1. Indeed, if we set $\gamma := \gamma_j$, then every coefficient z_i of the response $\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$ computed in the protocol gives rise to an ILWE sample

$$z_i = \langle \mathbf{c}_i, \mathbf{x}\rangle + y_i,$$

where $\mathbf{x}$ is a subkey of the secret $\mathbf{s}_1$ and $\mathbf{c}_i$ is derived from the challenge $c \in R$. There is, however, a crucial difference between ILWE and the ID protocol: The prover restarts the protocol whenever an ILWE sample z_i is outside the range $[\pm(\gamma_j - \beta)]$. In particular, in the ID protocol, an attacker never sees ILWE samples $z_i \notin [\pm(\gamma_j - \beta)]$. As shown by Lyubashevsky, this guarantees that no information on the secret key is revealed to an attacker. More formally, Lyubashevsky's proof shows that whenever an ILWE sample z_i is bounded in absolute value by $|z_i| \leq \gamma_j - \beta$, then the distribution of the tuple $(\mathbf{c}_i, z_i)$ is independent of $\mathbf{x}$. This motivates the following definition.

Definition 2. *(Informative ILWE samples)* Let z_i be an ILWE sample from Equation (1). We call z_i a *zero-knowledge sample* if $|z_i| \leq \gamma_j - \beta$, where $\beta := \eta\tau$, and we call z_i an *informative sample* otherwise.

While Lyubashevsky's proof was originally meant for proving security of the ID protocol, we will see below that it provides valuable insights for the cryptanalysis of ILWE. The main idea behind the proof is that zero-knowledge samples follow the uniform distribution over $[\pm(\gamma_j - \beta)]$. This, in turn, can be proven using the following lemma.

Lemma 3. *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 $[\pm\gamma_j]$, 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*

$$\Pr[z = x] = \frac{\Pr[x - \gamma_j \leq \langle \mathbf{c}, \mathbf{x} \rangle \leq x + \gamma_j]}{2\gamma_j + 1}.$$

Proof. By definition of z, we have

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

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

$$\Pr[z = x] = \frac{1}{2\gamma_j + 1} \cdot \sum_{i=x-\gamma_j}^{x+\gamma_j} \Pr[\langle \mathbf{c}, \mathbf{x} \rangle = i],$$

from which the desired statement immediately follows. □

For our ILWE instance with $|\langle \mathbf{c}_i, \mathbf{x} \rangle| \leq \beta$, Lemma 3 shows that the distribution of every ILWE sample $z_i = \langle \mathbf{c}_i, \mathbf{x} \rangle + y_i$ is

$$\Pr[z_i = x] = \frac{1}{2\gamma_j + 1} \cdot \begin{cases} 1, & \text{if } -\gamma_j + \beta \leq x \leq \gamma_j - \beta, \\ \Pr[\langle \mathbf{c}_i, \mathbf{x} \rangle \geq x - \gamma_j], & \text{if } \gamma_j - \beta < x \leq \gamma_j + \beta, \\ \Pr[\langle \mathbf{c}_i, \mathbf{x} \rangle \leq x + \gamma_j], & \text{if } -\gamma_j - \beta \leq x < -\gamma_j + \beta, \\ 0, & \text{else.} \end{cases} \tag{5}$$

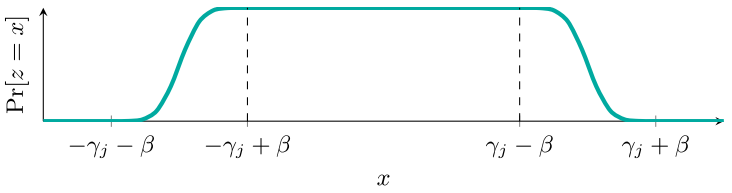

Fig. 3. Distribution of $z_i = \langle \mathbf{c}_i, \mathbf{x} \rangle + y_i$ with $|\langle \mathbf{c}_i, \mathbf{x} \rangle| \leq \beta$ and $y_i \leftarrow [\pm\gamma_j]$.

As illustrated in Fig. 3, the zero-knowledge samples $z_i \in [\pm(\gamma_j - \beta)]$ thus indeed follow the uniform distribution over the interval $[\pm(\gamma_j - \beta)]$. (To visualize both $\Pr[\langle \mathbf{c}_i, \mathbf{x} \rangle \geq x - \gamma_j]$ and $\Pr[\langle \mathbf{c}_i, \mathbf{x} \rangle \leq x + \gamma_j]$ in Fig. 3, we use the cumulative distribution function of a properly parameterized Gaussian distribution. This is justified by the central limit theorem, since $\langle \mathbf{c}_i, \mathbf{x} \rangle$ is the sum of τ uniformly random integers from $[\pm\eta]$.)

Note that zero-knowledge samples only exist if $\gamma_j > \beta$, i.e., if the inner products $\langle \mathbf{c}_i, \mathbf{x} \rangle$ are small compared to y_i. (Otherwise the region $-\gamma_j + \beta \leq x \leq \gamma_j - \beta$ in Fig. 3 is empty.) As we will see below, this is precisely the regime where we improve over the standard approach for solving ILWE.

4.2. *The Filter-and-Shift Preprocessing*

Since zero-knowledge samples do not reveal any information on the secret $\mathbf{x}$, they cannot help in solving the ILWE instance. Based on this observation, we propose with Algorithm 1 a novel preprocessing technique for ILWE.

Algorithm 1: filter-and-shift

Input : ILWE instance $(\mathbf{C}, \mathbf{z}) \in \mathbb{Z}^{m \times n} \times \mathbb{Z}^m$, parameters γ_j, β.
Output: Easier ILWE instance $(\mathbf{C}', \mathbf{z}') \in \mathbb{Z}^{m' \times n} \times \mathbb{Z}^{m'}$ with $m' \leq m$.

if $\gamma_j \leq \beta$ **then**
  return $(\mathbf{C}, \mathbf{z})$.
else
  Initialize empty matrix $\mathbf{C}'$ and empty vector $\mathbf{z}'$.
  for $i = 1, \ldots, m$ **do**
    if $\gamma_j - \beta < |z_i| \leq \gamma_j + \beta$ **then** ▷ Filter to informative, Def. 2.
      if $z_i > \gamma_j - \beta$ **then**
        $z_i \leftarrow z_i - \gamma_j + \beta$ ▷ Left shift.
      else if $-\gamma_j - \beta \geq z_i < -\gamma_j + \beta$ **then**
        $z_i \leftarrow z_i + \gamma_j - \beta$ ▷ Right shift.
      Append z_i to $\mathbf{z}'$ and $\mathbf{c}_i^T$ to $\mathbf{C}'$.
  return $(\mathbf{C}', \mathbf{z}')$.

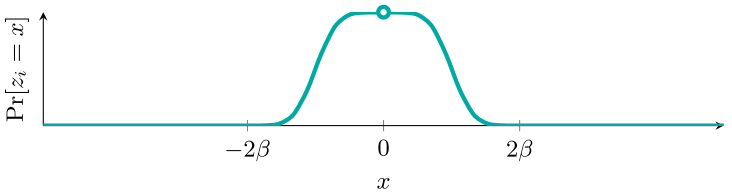

Fig. 4. Distribution of z_i after applying Equation (6). The circle at $x = 0$ is supposed to illustrate that $z = 0$ has probability 0.

Algorithm 1 *filters* out all zero-knowledge samples $z_i \in [\pm(\gamma_j - \beta)]$ from the ILWE instance, and transforms every informative sample z_i by *shifting* it as[1]

$$z_i \leftarrow \begin{cases} z_i - \gamma_j + \beta, & \text{if } z_i > \gamma_j - \beta, \\ z_i + \gamma_j - \beta, & \text{if } z_i < -\gamma_j + \beta. \end{cases} \tag{6}$$

Intuitively, the algorithm thus cuts out the region $-\gamma + \beta \leq x \leq \gamma - \beta$ from Fig. 3, thereby producing a new ILWE instance $(\mathbf{C}', \mathbf{z}')$ in which the samples follow the distribution shown in Fig. 4. As we will see below, $(\mathbf{C}', \mathbf{z}')$ is easier to solve for least squares regression than $(\mathbf{C}, \mathbf{z})$ is.

Error distribution. Comparing the distribution of the ILWE samples produced by Algorithm 1 (Fig. 4) with the distribution of the original samples (Fig. 3), it follows that the output $(\mathbf{C}', \mathbf{z}')$ of Algorithm 4 behaves like an ILWE instance in which the entries of the error $\mathbf{y}' := \mathbf{z}' - \mathbf{C}'\mathbf{x}$ follow the uniform distribution over $[\pm\beta]$. Indeed, if we set $\gamma_j := \beta$ in Fig. 3, then Figs. 3 and 4 become essentially identical. (The only insubstantial inaccuracy being here that $x = 0$ is excluded in Fig. 4.)

Number of ILWE samples. Suppose $\gamma_j \geq \beta$. Then by Equation (5), any given sample z_i in the original ILWE instance $(\mathbf{C}, \mathbf{z}) \in \mathbb{Z}^{m \times n} \times \mathbb{Z}^m$ is an informative sample with probability

$$\Pr[|z_i| > \gamma_j - \beta] = 1 - \sum_{x=-\gamma_j+\beta}^{\gamma_j-\beta} \Pr[z_i = x] = 1 - \frac{2(\gamma_j - \beta) + 1}{2\gamma_j + 1} = \frac{2\beta}{2\gamma_j + 1}. \tag{7}$$

The expected number of samples m' in the ILWE instance $(\mathbf{C}', \mathbf{z}') \in \mathbb{Z}^{m' \times n} \times \mathbb{Z}^{m'}$ produced by Algorithm 1 is therefore

$$\mathbb{E}[m'] = m \cdot \frac{2\beta}{2\gamma_j + 1} = \Theta\left(m \cdot \frac{\beta}{\gamma_j}\right). \tag{8}$$

Success condition for least squares regression. Recall from Equation (2) that the smallest m for which least squares regression solves the original ILWE instance $(\mathbf{C}, \mathbf{z}) \in$

[1] In its first **if** condition, Algorithm 1 furthemore checks whether $|z_i| \leq \gamma_j + \beta$. This may seem unnecessary for now, but will be useful later on.

$\mathbb{Z}^{m\times n} \times \mathbb{Z}^m$ (in which the error is uniform over $[\pm\gamma_j]$) grows as

$$m = \Theta\left(\frac{\gamma_j^2 n}{\tau}\log(n)\right).$$

As a consequence, for the ILWE instance $(\mathbf{C}', \mathbf{z}') \in \mathbb{Z}^{m'\times n} \times \mathbb{Z}^{m'}$ (in which the error is essentially uniform over $[\pm\beta]$) the smallest such m' grows as

$$m' = \Theta\left(\frac{\beta^2 n}{\tau}\log(n)\right).$$

Using Equation (8) to estimate m' and using $\beta = \eta\tau$, we rewrite the above as

$$m = \Theta\left(\gamma_j \eta n \log(n)\right).$$

This is precisely the desired bound from Equation (3), which improves over the state of the art whenever $\gamma_j > \beta$.

5. Relation Extraction

Bit leak breaks zero-knowledge. In the Leaky-Signature-LWE problem (Definition 1), we are given arbitrarily many relations $(\mathbf{c}, z, y_j) \in [\pm 1]_\tau^n \times \mathbb{Z} \times \{0, 1\}$, where z is defined as

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

for some secret error y, and y_j is the j-th bit of the two's complement of y. Grouping all z's, y's and $\mathbf{c}$'s into vectors $\mathbf{z}$, $\mathbf{y}$ and a matrix $\mathbf{C}$, respectively, gives rise to an ILWE instance

$$\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y}\,,$$

with solution $\mathbf{x}$. Importantly, since z is drawn according to the rejection sampling procedure of Fiat-Shamir with Aborts, the ILWE instance consists only of zero-knowledge samples (as defined in Sect. 4). Consequently, it is information-theoretically unsolvable. For instance, linear regression-based algorithms fail here, because the distribution of the error $\mathbf{y}$ does not meet the necessary requirements: Since $\mathbf{z}$ follows the uniform distribution over $[\pm(\gamma - \beta)]^m$, where m is the number of ILWE samples, it follows that the distribution of $\mathbf{y} = \mathbf{z} - \mathbf{C}\mathbf{x}$ conditioned on $\mathbf{C}\mathbf{x}$ is the uniform distribution over the interval

$$I_{\mathbf{C}\mathbf{x}} := [\pm(\gamma - \beta)]^m - \mathbf{C}\mathbf{x}\,,$$

thereby failing to meet the requirements for linear regression to succeed [5].

As we will see below, having access to the leaked bit y_j allows us to efficiently transform every $z = \langle \mathbf{c}, \mathbf{x}\rangle + y$ into

$$\overline{z} := \langle \mathbf{c}, \mathbf{x}\rangle + [y]_{2^j}\,. \tag{9}$$

We call the process of computing $\overline{z}$ *relation extraction*. Grouping all $\overline{z}$'s into a vector $\overline{\mathbf{z}}$, we obtain a new ILWE instance

$$\overline{\mathbf{z}} = \mathbf{C}\mathbf{x} + [\mathbf{y}]_{2^j} .$$

The crucial observation is now that replacing the error $\mathbf{y} \leftarrow I_{\mathbf{C}\mathbf{x}}$ by $[\mathbf{y}]_{2^j}$ makes the error distribution close to the uniform distribution over $[\pm 2^{j-1}]^m$. As a result, if m is sufficently large, then the resulting ILWE instance does meet the requirements for linear regression to succeed: If we combine linear regression with our novel filter-and-shift technique from Sect. 4, then Equation (3) shows that we can solve the ILWE instance provided that

$$m = \Theta\left(2^j \eta n \log(n)\right) ,$$

where $\eta := \|\mathbf{x}\|_\infty$ and n is the dimension of $\mathbf{x}$. In other words, knowing one bit per entry of $\mathbf{y}$ allows us to break zero-knowledge, and thereby to recover the secret $\mathbf{x}$.

Differences with [36]. The seminal randomness leakage attack of Liu, Zhou, Sun, Wang, Zhang, and Ming (LZS$^+$) also uses leakage in $\mathbf{y}$ to transform the unsolvable ILWE instance $\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y}$ into a solvable instance, in which the transformed error is obtained by performing modular reduction on the original error $\mathbf{y}$. There are, however, two differences from the approach outlined above. First, while we work with the two's complement, LZS$^+$ work with the less standard *sign-and-magnitude* bit representation. Second, while we work with the standard $[\cdot]_{2^j}$ modular reduction, LZS$^+$ reduce $y \in \mathbb{Z}$ to some value $\widetilde{y}$ defined as follows:

1. If $y \geq 0$, then $\widetilde{y}$ is the unique value in $[0, 2^j)$ satisfying $\widetilde{y} \equiv y \mod 2^j$.
2. If $y < 0$, then $\widetilde{y}$ is the unique value in $(-2^j, 0)$ satisfying $\overline{y} \equiv y \mod 2^j$.

LZS$^+$ thus performs some kind of sign-preserving modular reduction.

As a result, LZS$^+$ obtain an ILWE instance in which the error is close to uniform over $[\pm 2^j]$, whereas in ours it is close to uniform over $[\pm 2^{j-1}]$. In particular, our ILWE instance is slightly easier to solve. Another advantage of our approach is that our relation extraction from $z = \langle \mathbf{c}, \mathbf{x} \rangle + y$ to $\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}$ is provably correct, whereas LZS$^+$ only claim that theirs is "almost always correct". Furthermore, in contrast to ours, the approach of LZS$^+$ can not handle the particularly interesting (and arguably practically highly relevant) case where the leaked bit y_j corresponds to the sign of y.

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

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

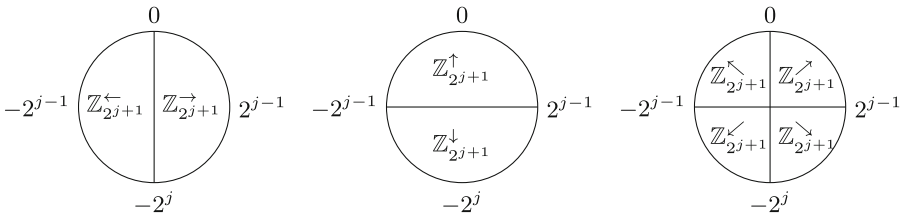

Fig. 5. Intuition for our partitions of $\mathbb{Z}_{2^j+1}$.

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 Fig. 5. 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 Fig. 5, to prove two technical lemmas.

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

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

Proof. Looking at Fig. 5, 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 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 two's complement, we have $x = \left[\sum_{i=0}^{w-1} x_i 2^i\right]_{2^w}$. Together with Fig. 5, this proves the lemma. □

With the lemmas above, we are now ready to describe the relation extraction for producing $\overline{z}$ as in Equation (9).

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 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

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

and

$$b_j := \begin{cases} y_j, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}},\\ y_j \oplus 1, & \text{if } [z]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}},\end{cases} \tag{11}$$

respectively.

Let us explain the effect of the pre-processing. In modulo-2^{j+1} arithmetic, adding 2^{j-1} corresponds to rotating the circles from Fig. 5 by 90 degrees clockwise. Similarly, subtracting 2^{j-1} corresponds to rotating the circle by 90 degrees counter clockwise. Hence, Equation (10) ensures that

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}\,. \tag{12}$$

Let us write $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, for some unknown $y^{\uparrow}$, which (by Equation (10)) 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 (11), Lemma 5 and the fact that adding 2^{j-1} rotates the circles Fig. 5 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

$$b_j = 1 \iff [y^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}} . \tag{13}$$

Relations $(\mathbf{c}, z^{\uparrow}, b_j)$ with $z^{\uparrow} = \langle \mathbf{c}, \mathbf{x} \rangle + y^{\uparrow}$, for which both Equations (12) and (13) hold, are called *normal form relations*.

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

$$\overline{z} := \begin{cases} [z^{\uparrow}]_{2^{j+1}}, & \text{if } b_j = 1, \\ [z^{\uparrow}]_{2^{j+1}} + 2^j, & \text{if } b_j = 0 \text{ and } [z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}, \\ [z^{\uparrow}]_{2^{j+1}} - 2^j, & \text{if } b_j = 0 \text{ and } [z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}. \end{cases} \tag{14}$$

As we show in Theorem 6 below, the resulting $\overline{z}$ has the desired shape. Worth noting, as in the original relation extraction of [36], 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 (14) 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

$$u \equiv y^{\uparrow} \mod 2^{j+1} , \tag{15}$$

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

$$\overline{z} = \langle \mathbf{c}, \mathbf{x} \rangle + [u]_{2^j} . \tag{16}$$

Since $(\mathbf{c}, z^{\uparrow}, b_j)$ is a normal form relation, we have by Equation (12) that $[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\uparrow}_{2^{j+1}}$. Let us distinguish the two cases

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\nwarrow}_{2^{j+1}}\,, \tag{17}$$

and

$$[z^{\uparrow}]_{2^{j+1}} \in \mathbb{Z}^{\nearrow}_{2^{j+1}}\,. \tag{18}$$

Suppose we are in the case of Equation (17). Then $[z^{\uparrow}]_{2^{j+1}}$ lies in the upper left quarter of the circles from Fig. 6, 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

$$-2^{j} \leq u < 2^{j-1}\,. \tag{19}$$

In other words, u either lies in the upper left quarter, the upper right quarter, or the lower left quarter of the circles from Fig. 6, i.e.,

$$u \notin \mathbb{Z}^{\searrow}_{2^{j+1}}. \tag{20}$$

Combining Equations (15) and (19), 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 (13) 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 (20),

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

If we are instead in the case of Equation (18), 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 (14) and the fact that $[z^{\uparrow}]_{2^{j+1}} = \langle \mathbf{c}, \mathbf{x}\rangle + u$, this implies Equation (16) and thus concludes the proof of the theorem. □

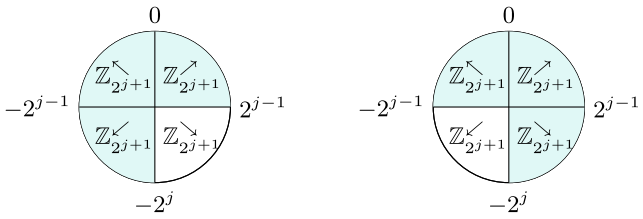

Fig. 6. Intuition for the proof of Theorem 6. The colored area is the range for $u := [z^{\uparrow}]_{2^j+1} - \langle \mathbf{c}, \mathbf{x} \rangle$, depending on whether $[z^{\uparrow}]_{2^j+1} \in \mathbb{Z}^{\nwarrow}_{2^j+1}$ (left) or $[z^{\uparrow}]_{2^j+1} \in \mathbb{Z}^{\nearrow}_{2^j+1}$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

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 Sect. 8, 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 $\sigma^2 = \frac{\eta(\eta+1)}{3} \cdot \tau = 2 \cdot 39 = 78$. (Recall that $\frac{\eta(\eta+1)}{3}$ is the variance of the discrete uniform distribution over $[\pm\eta]$.) It is well-known that a Gaussian random variable with variance σ^2 rarely 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.

Putting things together. The complete relation extraction is given in Algorithm 2. The full randomness leakage would now proceed as follows:

1. Run relation extraction (Algorithm 2) on input the leaked data $(\mathbf{c}, \mathbf{z}, \mathbf{y}_j)$ to obtain a solvable ILWE instance $\overline{\mathbf{z}} = \mathbf{C}\mathbf{x} + [\mathbf{y}]_{2^j}$.
2. Run filter-and-shift (Algorithm 1) on input $(\mathbf{C}, \overline{\mathbf{z}})$ with parameters β and $\gamma_j := 2^{j-1}$ to obtain an easier ILWE instance.
3. Solve the resulting ILWE instance using least squares regression.

However, the above attack would only succeed if the leaked bits y_j are always correct. Below, we show how to adapt the attack to a more realistic setting in which the leak is noisy and flips each y_j with some error rate p. In a nutshell, this requires us to enhance the above approach by incorporating a *scaling step* into least squares regression.

Algorithm 2: Relation Extraction

Input : List L of relations $(\mathbf{c}, z, y_j)$ with
challenge vector $\mathbf{c} \in [\pm 1]^n_\tau$,
signature coefficient $z \in [\pm(\gamma - \beta)]$,
randomness bit $y_j \in [0, 1]$,
leakage index j.
Output: An ILWE instance $(\mathbf{C}, \mathbf{z})$.

Initialize empty matrix $\mathbf{C}$ and vector $\overline{\mathbf{z}}$.
for $(\mathbf{c}, z, y_j) \in L$ **do**
  Append $\mathbf{c}$ to $\mathbf{C}$.
  if $[z]_{2j+1} \in \mathbb{Z}^{\leftarrow}_{2j+1}$ **then** ▷ Normal form computation, Eq.(10)&(11).
    $z^{\uparrow} := z + 2^{j-1}$
    $b_j := y_j$
  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.(14).
    $\overline{z} := [z^{\uparrow}]_{2j+1}$
  else if $[z^{\uparrow}]_{2j+1} \in \mathbb{Z}^{\leftarrow}_{2j+1}$ **then**
    $\overline{z} := [z^{\uparrow}]_{2j+1} + 2^j$
  else
    $\overline{z} := [z^{\uparrow}]_{2j+1} - 2^j$
  Append $\overline{z}$ to $\mathbf{z}$.
return $(\mathbf{C}, \mathbf{z})$.

6. Our Attack in the Presence of Noise

The leak bit y_j may be corrupted by side-channel noise, such that it flips according to some *error rate*.

Definition 7. *(Error rate)* Let $(\mathbf{c}, z, y_j^{(p)})$ be a tuple from Definition 1, where the leaked bit $y_j^{(p)}$ is obtained by independently flipping the true bit y_j with probability $p \in [0, 0.5)$, i.e.,

$$y_j^{(p)} = y_j \oplus f_j\,, \qquad f_j \sim \text{Bernoulli}(p)\,.$$

We refer to p as the *error rate*.

Remark 8. Any error rate $p > 0.5$ can be reduced to an error rate $p < 0.5$ by flipping all leak bits. Hence, we restrict to $p < 0.5$.

For an error rate p, the resulting sample distribution is a p-weighted mixture of the extreme case where the leak bit is always correct and the opposite extreme where the leak bit is always flipped. In such a setting, we propose to solve the ILWE instance obtained from relation extraction and filter-and-shift using our novel Algorithm 3, which enhances the standard approach based on least squares regression by a scaling step in lines 2 and 3 of Algorithm 3. Our complete randomness leakage attack is provided in Algorithm 4.

Algorithm 3: Subkey Estimation

Input : $(\mathbf{C}, \mathbf{z}) \in \mathbb{Z}^{m \times n} \times \mathbb{Z}^m$,
error rate $p \in [0, 0.5)$.
Output: LWE secret $\mathbf{s}_1 \in ([\pm\eta]^n)^\ell$.

1 Apply 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 subkey $\mathbf{x}$.
2 Set scaling factor $r = (1 - 2p)^{-1}$.
3 Compute the subkey $\mathbf{x} := \lfloor r \cdot \hat{\mathbf{x}} \rceil \in \mathbb{Z}^n$ of $\mathbf{s}_1$ via coordinate-wise rounding.
4 **return** $\mathbf{x}$.

Algorithm 4: Leaky-Signature-LWE Attack

Input : List L of tuples $(\mathbf{c}, z, y_j^{(p)})$ with
challenge vector $\mathbf{c} \in [\pm 1]_\tau^n$,
signature coefficient $z \in [\pm(\gamma - \beta)]$,
randomness bit $y_j^{(p)} \in [0, 1]$,
leakage index j,
public key $(\mathbf{A}, \mathbf{t})$,
error rate $p \in [0, 0.5)$. ; ▷ `Definition 7, p = 0 noise-free.`
Output: LWE secret $\mathbf{s}_1 \in ([\pm\eta]^n)^\ell$.

1 $(\mathbf{C}, \mathbf{z}) :=$ `RelationExtraction`(L, j) ; ▷ `Algorithm 2.`
2 $(\mathbf{C}', \mathbf{z}') :=$ `FilterAndShift`$(\mathbf{C}, \mathbf{z}, 2^{j-1}, \beta)$ ▷ `Algorithm 1.`
3 $\mathbf{x} :=$ `SubKeyEstimation`$(\mathbf{C}', \mathbf{z}', p)$ ▷ `Algorithm 3.`
4 Recover the full LWE secret $\mathbf{s}_1$ via lattice reduction using the public key $(\mathbf{A}, \mathbf{t})$ and $\mathbf{x}$. ▷ `See Section 3.1.`
5 **return** $\mathbf{s}_1$.

Let us analyze how tuples $(\mathbf{c}, z, y_j^{(p)})$ with a flipped leak bit are processed by the relation extraction (Algorithm 2) and the subsequent call to filter-and-shift (Algorithm 1) inside Algorithm 4.

Relation extraction. We first notice that the leak bit is used only in the normal form computation of Algorithm 2 and not in any subsequent steps. Therefore, it suffices to determine the output of the normal form computation for a flipped leak bit and to analyze how this output is handled by subsequent steps. Let us denote the flipped leak bit by $y_j^{(f)} := y_j^{(c)} \oplus 1$, where $y_j^{(c)}$ denotes the true bit. When initialized with $y_j^{(f)}$, the normal form computation (Equation (11)) results in a flipped indicator bit

$$b_j^{(f)} = 1 \oplus b_j^{(c)} .$$

The subsequent step, i.e., the relation extraction, conditionally adds or subtracts 2^j, depending on the value of this indicator bit.

We evaluate the relation extraction in the case of a flipped leak bit against the case of a correct leak bit.

Theorem 9. (Relation extraction with a flipped leak bit) *Let* $\overline{z}^{(f)}$ *and* $\overline{z}^{(c)}$ *be defined as in Equation* (14) *using a flipped and a correct leak bit, respectively. Then,*

$$\overline{z}^{(f)} = \overline{z}^{(c)} - \operatorname{sign}\left(\overline{z}^{(c)}\right) 2^j \,.$$

Proof. Recall $\mathbb{Z}^{\leftarrow}_{2^{j+1}} = \{-2^j, -2^j + 1, \ldots, -1\}$ and $\mathbb{Z}^{\rightarrow}_{2^{j+1}} = \mathbb{Z} \setminus \mathbb{Z}^{\leftarrow}_{2^{j+1}}$ and, for $z \in [\pm 2^j]$, the relation extraction

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

We have $b_j^{(f)} = 1 \oplus b_j^{(c)}$. Let us distinguish the cases for $b_j^{(f)}$ and for $b_j^{(c)}$.
Case 1: $b_j^{(f)} = 1$. If $z \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$, the addition of 2^j that occurs for $b_j^{(c)} = 0$ is omitted, which gives

$$\overline{z}^{(f)} = \overline{z}^{(c)} - 2^j \,,$$

where $\overline{z}^{(c)} \geq 0$. If $z \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}$, the subtraction of 2^j is omitted, and we obtain

$$\overline{z}^{(f)} = \overline{z}^{(c)} + 2^j \,,$$

where $\overline{z}^{(c)} < 0$.
Case 2: $b_j^{(f)} = 0$. If $z \in \mathbb{Z}^{\leftarrow}_{2^{j+1}}$, then 2^j is added, which gives

$$\overline{z}^{(f)} = \overline{z}^{(c)} + 2^j \,,$$

where $\overline{z}^{(c)} < 0$. If $z \in \mathbb{Z}^{\rightarrow}_{2^{j+1}}$, then 2^j is subtracted, resulting in

$$\overline{z}^{(f)} = \overline{z}^{(c)} - 2^j \,,$$

where $\overline{z}^{(c)} \geq 0$.

In all cases, flipping the leak bit shifts $\overline{z}$ by exactly 2^j in the direction opposite to its sign. This yields

$$\overline{z}^{(f)} = \overline{z}^{(c)} - \operatorname{sign}\left(\overline{z}^{(c)}\right) 2^j \,,$$

as claimed. □

Recall that $\overline{z}^{(c)}$ is of the form $\overline{z}^{(c)} = \langle \mathbf{c}, \mathbf{x} \rangle + [y]_{2^j}$, and therefore follows the distribution shown in Fig. 3 (with $\gamma_j := 2^{j-1}$). Theorem 9 therefore shows that the resulting distribution of $\overline{z}^{(f)}$ is the top graph in Fig. 7.

Filter-and-shift. We now consider how filter-and-shift (Algorithm 1) acts on $\overline{z}^{(f)}$. Recall that filter-and-shift (instantiated with $\gamma_j := 2^{j-1}$) filters out all ILWE samples that do not lie in between the dotted bars in Fig. 7, i.e., the algorithm removes all $\overline{z}^{(f)}$ satisfying

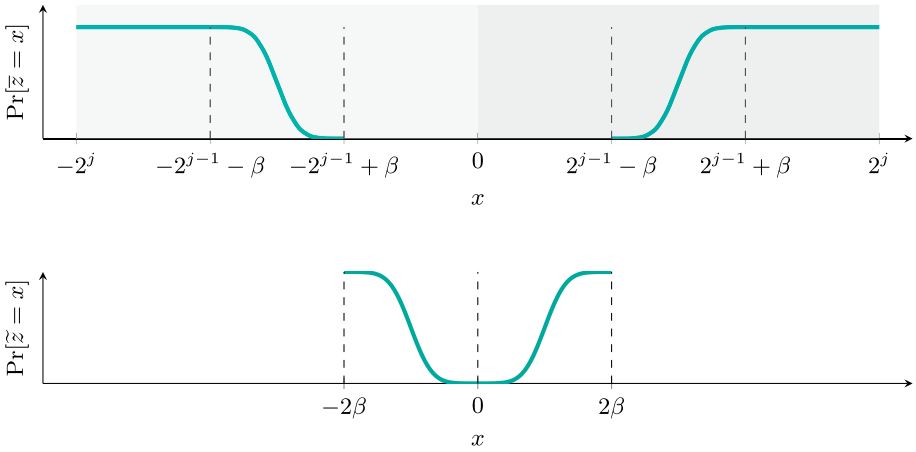

Fig. 7. The top graph shows the distribution of $\overline{z}^{(f)} = \overline{z}^{(c)} - \mathrm{sign}(\overline{z}^{(c)}) \cdot 2^j$. The bottom graph shows the distribution of $\overline{z}^{(f)}$ after applying filter-and-shift.

1. $\overline{z}^{(f)} < -2^{j-1} - \beta$,
2. $\overline{z}^{(f)} > 2^{j-1} + \beta$, or
3. $|\overline{z}^{(f)}| \leq 2^{j-1} - \beta$.

The remaining $\overline{z}^{(f)}$ are then shifted and thereby result in the distribution shown in the bottom graph of Fig. 7.

Comparing Fig. 7 with the distribution of $\overline{z}^{(c)}$ (recall Fig. 4), it follows that, after applying filter-and-shift, $\overline{z}^{(f)}$ is given by

$$\overline{z}^{(f)} = \overline{z}^{(c)} - 2\beta \,\mathrm{sign}(\overline{z}^{(c)}) \,.$$

In the presence of bit flips of errors, we thus end up with an ILWE instance in which the samples can be modeled via the following distribution.

Definition 10. (Noisy ILWE) Fix some public parameters η, τ, and β. In the Noisy ILWE problem, we are given $\mathbf{C} \in [\pm 1]_\tau^{m \times n}$ and $\mathbf{z} \in [\pm\eta]^n$ such that

$$\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y} + \mathbf{e}$$

where each entry of $\mathbf{y}$ is drawn uniformly at random from $[\pm\beta]$, and $\mathbf{e}$ is defined as

$$\mathbf{e} := -2\beta\mathbf{f} \odot \mathrm{sign}(\mathbf{C}\mathbf{x} + \mathbf{y}) \,,$$

for some $\mathbf{f} \sim \mathrm{Bernoulli}(p)^m$, $\odot$ as element-wise product, and $\mathrm{sign}(\cdot)$ is applied entry wise to $\mathbf{C}\mathbf{x} + \mathbf{y}$. The goal is to recover $\mathbf{x}$.

We show below that the call to Algorithm 3 inside Algorithm 4 successfully solves Noisy ILWE, provided that the number of samples m is sufficiently large.

7. Least Squares Regression for Solving Noisy ILWE

To prove that Algorithm 3 recovers the secret subkey $\mathbf{x}$, we proceed in three steps. First, we show in Sect. 7.1 that the least squares estimator

$$\hat{\mathbf{x}} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z} \tag{21}$$

from Algorithm 3, Line 1 exists with high probability and that its expectation is a multiple of the secret subkey, such that it can be rescaled (Algorithm 3, Line 2). We then show in Sect. 7.2 that this estimator converges to its expectation.

Finally, in Sect. 7.3, we derive an estimate of the sample complexity by applying a central limit theorem, and subsequently a Gaussian tail bound. We ensure that the estimation error in each component after rescaling does not exceed $\frac{1}{2}$, so that rounding (Algorithm 3, Line 3) correctly recovers the secret subkey.

7.1. *Expectation of Least Squares Estimator*

Let us first establish that $\mathbf{C}^T\mathbf{C}$ is invertible with high probability, which implies the existence of the least squares estimator. In the same lemma, we derive the expectation of $\frac{1}{m}\mathbf{C}^T\mathbf{C}$, which is needed to compute the expectation of the least squares estimator.

Lemma 11. *Let* $\mathbf{C}$ *be an* $(m \times n)$ *matrix with rows independently sampled from* $[\pm 1]_\tau^n$. *For* $m > n$, *it holds that*

$$\mathbb{E}_{\mathbf{C}}\left[\frac{1}{m}\mathbf{C}^T\mathbf{C}\right] = \frac{\tau}{n}\mathbf{I}_n\,. \tag{22}$$

Each entry of $\frac{1}{m}\mathbf{C}^T\mathbf{C}$ *converges to its expected value* 0 *(off-diagonal) or* $\frac{\tau}{n}$ *(on-diagonal) exponentially fast in* m. *For* $m > 2n\ln(n)$, *it holds that*

$$\Pr[\,\mathbf{C}^T\mathbf{C} \textit{ is invertible}\,] > 1 - ne^{-\frac{m}{2n}}\,. \tag{23}$$

Proof. See Appendix A. □

As a consequence of Lemma 11, the least squares estimator exists with probability close to one for sufficiently large m. In the following, we compute its expectation for error rates $p \in [0, 0.5)$.

Theorem 12. (Expectation of least squares estimator) *Consider the OLS estimate* $\hat{\mathbf{x}} = (\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{z}$, *where* $\mathbf{C}$ *and* $\mathbf{z}$ *are defined as in Definition 10, for some error rate* $p \in [0, 0.5)$. *Then,*

$$\mathbb{E}[\hat{\mathbf{x}}] = (1 - 2p)\mathbf{x}\,.$$

Proof. Write the sample vector as

$$\mathbf{z} = \mathbf{C}\mathbf{x} + \mathbf{y} + \mathbf{e}\,, \qquad e_i := -2\beta f_i \operatorname{sign}(\langle \mathbf{c}_i, \mathbf{x}\rangle + y_i)\,.$$

The least squares estimator satisfies

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

By Lemma 11, $\mathbf{C}^T\mathbf{C}$ is invertible with high probability such that we can compute the estimator. Taking expectations and using linearity,

$$\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}[\hat{\mathbf{x}}] = \mathbf{x} + \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{y}\right] + \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{e}\right].$$

The second term vanishes since $\mathbf{y}$ has zero mean and is independent of $\mathbf{C}$.
To evaluate the third term, we compute

$$\begin{aligned}\mathbb{E}_y[\operatorname{sign}(\langle\mathbf{c},\mathbf{x}\rangle + y)] &= (-1)\Pr[\langle\mathbf{c},\mathbf{x}\rangle + y < 0] + (+1)\Pr[\langle\mathbf{c},\mathbf{x}\rangle + y \geq 0]\\ &= -\Pr[y < -\langle\mathbf{c},\mathbf{x}\rangle] + \Pr[y \geq -\langle\mathbf{c},\mathbf{x}\rangle]\\ &= -\frac{\beta - \langle\mathbf{c},\mathbf{x}\rangle}{2\beta} + \frac{\beta + \langle\mathbf{c},\mathbf{x}\rangle}{2\beta}\\ &= \frac{\langle\mathbf{c},\mathbf{x}\rangle}{\beta}\,,\end{aligned}$$

where the third equality follows from the fact that y is uniform on $[\pm\beta]$ and $|\langle\mathbf{c},\mathbf{x}\rangle| < \beta$. This computation together with independence of f_i, y_i, and $\mathbf{c}_i$ yields

$$\begin{aligned}\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{e}\right] &= -2\beta\,\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\begin{pmatrix} f_1\ \operatorname{sign}(\langle\mathbf{c}_1,\mathbf{x}\rangle + y_1)\\ f_2\ \operatorname{sign}(\langle\mathbf{c}_2,\mathbf{x}\rangle + y_2)\\ \vdots\\ f_m\ \operatorname{sign}(\langle\mathbf{c}_m,\mathbf{x}\rangle + y_m)\end{pmatrix}\right]\\ &= -\frac{2p\beta}{\beta}\,\mathbb{E}_{\mathbf{C}}\left[(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\begin{pmatrix}\langle\mathbf{c}_1,\mathbf{x}\rangle\\ \langle\mathbf{c}_2,\mathbf{x}\rangle\\ \vdots\\ \langle\mathbf{c}_m,\mathbf{x}\rangle\end{pmatrix}\right]\\ &= -2p\,\mathbf{x}\,. \qquad (24)\end{aligned}$$

Combining terms gives

$$\mathbb{E}[\hat{\mathbf{x}}] = (1-2p)\mathbf{x}\,,$$

as claimed. □

Theorem 12 implies that, in expectation, for $p = 0$, the estimator recovers $\mathbf{x}$. For $p = \frac{1}{2}$, the expectation vanishes, making recovery impossible (which is intuitive, as now leak bits are perfectly random). In the presence of noise, the least squares estimator has to be *rescaled* by $r = \frac{1}{1-2p}$, as done in Line 2 in Algorithm 3.

That is, for $p \in [0, \frac{1}{2})$, the *scaled* least squares estimator $r\hat{\mathbf{x}}$ is an unbiased estimator of the true $\mathbf{x}$:

$$\mathbb{E}[r\hat{\mathbf{x}}] = \frac{1-2p}{1-2p}\mathbf{x} = \mathbf{x}\,. \tag{25}$$

7.2. *Convergence of Least Squares Estimator*

To establish convergence of the least squares estimator to its expected value, we show that its covariance matrix vanishes as the number of samples increases. This requires deriving the expectation of the inverse Gram matrix $(\mathbf{C}^T\mathbf{C})^{-1}$.

As the following lemma invokes a matrix norm, we provide a (very) brief reminder: Recall that the spectral norm $\|\cdot\|_2$ of a matrix equals its largest singular value. For symmetric matrices, including the Gram matrix, this simplifies to the largest absolute eigenvalue.

Lemma 13. (Approximation of the inverse gram matrix) *Let* $\mathbf{C}$ *be an* $(m \times n)$ *matrix with rows independently sampled from* $[\pm 1]^n_\tau$. *For the matrix* $\mathbf{C}^T\mathbf{C}$ *and* m *large enough, its inverse is given as*

$$\mathbb{E}_{\mathbf{C}}\left[\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\right] = \frac{n}{\tau m}\mathbf{I}_n + \mathcal{O}(\tfrac{1}{m} \parallel \boldsymbol{\Delta} \parallel^2)\,, \tag{26}$$

where $\boldsymbol{\Delta} = \frac{1}{m}\mathbf{C}^T\mathbf{C} - \frac{1}{m}\mathbb{E}[\mathbf{C}^T\mathbf{C}]$ *is the observation error of* $\frac{1}{m}\mathbf{C}^T\mathbf{C}$ *(which decays exponentially fast with* m *according to Lemma 11).*

Proof. To show this lemma we make use of the Neumann Series to approximate the inverse of a slightly perturbed matrix [49]. Assume that we only observe a matrix $\mathbf{A}$ up to some small perturbation matrix $\boldsymbol{\Delta}$, i.e.,

$$\mathbf{A} = \mathbb{E}[\mathbf{A}] + \boldsymbol{\Delta}\,.$$

Then, we can approximate its inverse, $\mathbf{A}^{-1}$, as

$$\mathbf{A}^{-1} = (\mathbb{E}[\mathbf{A}] + \boldsymbol{\Delta})^{-1} = \mathbb{E}[\mathbf{A}]^{-1} - \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta}\mathbb{E}[\mathbf{A}]^{-1} + \mathcal{O}(\|\boldsymbol{\Delta}\|^2)\,. \tag{27}$$

To see this, we start with

$$\begin{aligned}\mathbf{A}^{-1} = (\mathbb{E}[\mathbf{A}] + \boldsymbol{\Delta})^{-1} &= \Big(\mathbb{E}[\mathbf{A}](\mathbf{I} + \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta})\Big)^{-1} \\ &= \big(\underbrace{\mathbf{I} + \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta}}_{\mathbf{B}}\big)^{-1}\mathbb{E}[\mathbf{A}]^{-1}\,.\end{aligned}$$

For the matrix $\mathbf{B} = \mathbf{I} + \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta}$, we now assume that it's Neumann Series [49, Theorem 4.20] converges, i.e.,

$$\lim_{k\to\infty}(I - \mathbf{B})^k = \lim_{k\to\infty}(-\mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta})^k = 0\,. \tag{28}$$

Provided that Equation (28) holds, it directly follows that:

$$\mathbf{B}^{-1} = \sum_{k=0}^{\infty}(I-\mathbf{B})^k = \sum_{k=0}^{\infty}(-\mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta})^k = \mathbf{I} - \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta} + \mathcal{O}(\|\boldsymbol{\Delta}\|^2)\,,$$

and therefore

$$\begin{aligned}\mathbf{A}^{-1} &= \left(\mathbf{I} + \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta}\right)^{-1}\mathbb{E}[\mathbf{A}]^{-1} = \left(\sum_{k=0}^{\infty}(-\mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta})^k\right)\mathbb{E}[\mathbf{A}]^{-1} \\ &= \mathbb{E}[\mathbf{A}]^{-1} - \mathbb{E}[\mathbf{A}]^{-1}\boldsymbol{\Delta}\mathbb{E}[\mathbf{A}]^{-1} + \mathcal{O}(\|\boldsymbol{\Delta}\|^2)\,.\end{aligned}$$

as claimed in Equation (27).

Coming back to our setting, we can view the observed Gram matrix, $\frac{1}{m}\mathbf{C}^T\mathbf{C}$, (with finitely many samples) as a disturbed version of its expected value

$$\frac{1}{m}\mathbf{C}^T\mathbf{C} = \frac{1}{m}\mathbb{E}[\mathbf{C}^T\mathbf{C}] + \boldsymbol{\Delta}\,.$$

We already know from Lemma 11 that

$$\mathbb{E}\left[\frac{1}{m}\mathbf{C}^T\mathbf{C}\right] = \frac{\tau}{n}\mathbf{I}_n \implies \mathbb{E}\left[\frac{1}{m}\mathbf{C}^T\mathbf{C}\right]^{-1} = \frac{n}{\tau}\mathbf{I}_n\,, \quad \mathbb{E}[\boldsymbol{\Delta}] = 0\,,$$

and that the matrix $\boldsymbol{\Delta}$ is small as its entries decay exponentially fast with m. We proceed to verify that the Neumann Series actually converges, i.e., whether Equation (28) holds (for $\mathbf{A} = \frac{1}{m}\mathbf{C}^T\mathbf{C}$), and we find that indeed

$$\lim_{k\to\infty}(\mathbb{E}[\tfrac{1}{m}\mathbf{C}^T\mathbf{C}]^{-1}\boldsymbol{\Delta})^k = \lim_{k\to\infty}(\tfrac{n}{\tau}\boldsymbol{\Delta})^k = 0\,.$$

The last equation follows from the fact that $\|\boldsymbol{\Delta}\|_2 \to 0$ with m (Lemma 11), which implies $\|\frac{n}{\tau}\boldsymbol{\Delta}\|_2 < 1$ for some m large enough (i.e., the magnitude of all eigenvalues of $\frac{n}{\tau}\boldsymbol{\Delta}$ is smaller than one, which is the sufficient condition required in [49, Theorem 4.20]). That is, Equation (27) applies for $\mathbf{A} = \frac{1}{m}\mathbf{C}^T\mathbf{C}$ and we obtain

$$\left(\tfrac{1}{m}\mathbf{C}^T\mathbf{C}\right)^{-1} = \mathbb{E}[\left(\tfrac{1}{m}\mathbf{C}^T\mathbf{C}\right)]^{-1} - \mathbb{E}[\left(\tfrac{1}{m}\mathbf{C}^T\mathbf{C}\right)]^{-1}\boldsymbol{\Delta}\mathbb{E}[\left(\tfrac{1}{m}\mathbf{C}^T\mathbf{C}\right)]^{-1} + \mathcal{O}(\|\boldsymbol{\Delta}\|^2)\,. \quad (29)$$

Lastly, we take the expectation (invoking that $\mathbb{E}[\boldsymbol{\Delta}] = 0$) and obtain

$$\mathbb{E}_{\mathbf{C}}\left[\left(\tfrac{1}{m}\mathbf{C}^T\mathbf{C}\right)^{-1}\right] = \frac{n}{\tau}\mathbf{I}_n + \mathcal{O}(\|\ \boldsymbol{\Delta}\ \|^2)\,.$$

By linearity of expectation (and the fact that $(K\mathbf{A})^{-1} = \frac{1}{K}\mathbf{A}^{-1}$ for an invertible matrix $\mathbf{A}$ and $K \neq 0$), we obtain the approximation in Equation (26). □

By Lemma 13, the inverse of the Gram matrix is, in expectation, a scaled diagonal matrix plus an error term that tends to zero with increasing m. Neglecting this quickly vanishing error term, we can approximate the covariance matrix of the biased least squares estimator defined in Equation (21).

Estimate 14 (Covariance of least squares estimator). *Let* $\hat{\mathbf{x}} = (\mathbf{C}^\top \mathbf{C})^{-1}\mathbf{C}^\top \mathbf{z}$, *where* $\mathbf{C}$ *and* $\mathbf{z}$ *are defined as in Definition 10, for an error rate* $p \in [0, 0.5)$ *and with* $\mathrm{Var}[y_i] = \sigma_y^2$. *Then,*

$$\mathrm{Cov}[\hat{\mathbf{x}}] \approx \frac{1}{m}\left(\frac{n\sigma_y^2}{\tau} + 2p\left(\frac{n\beta^2}{\tau} + \|\mathbf{x}\|_2^2\right)\right)\mathbf{I}_n\,.$$

Justification. For brevity of notation, we write

$$\mathbf{C}^\dagger := (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\,.$$

Let $\mathbf{e} = (e_1, \ldots, e_n)$, where $e_i := -2\beta f_i \operatorname{sign}(\langle \mathbf{c}_i, \mathbf{x}\rangle + y_i)$. We further assume the key $\mathbf{x}$ to be fixed such that $\mathbf{x}$ is a constant. Then, the covariance can be computed as

$$\begin{aligned}
\mathrm{Cov}[\hat{\mathbf{x}}] &= \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}[(\hat{\mathbf{x}} - \mathbb{E}[\hat{\mathbf{x}}])(\hat{\mathbf{x}} - \mathbb{E}[\hat{\mathbf{x}}])^T] \\
&= \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\big((\mathbf{x} + \mathbf{C}^\dagger\mathbf{y} + \mathbf{C}^\dagger\mathbf{e}) - (1-2p)\mathbf{x}\big)\big((\mathbf{x} + \mathbf{C}^\dagger\mathbf{y} + \mathbf{C}^\dagger\mathbf{e} - (1-2p)\mathbf{x}\big)^T\Big] \\
&= \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\big(\mathbf{C}^\dagger\mathbf{y} + \mathbf{C}^\dagger\mathbf{e} + 2p\mathbf{x}\big)\big(\mathbf{C}^\dagger\mathbf{y} + \mathbf{C}^\dagger\mathbf{e} + 2p\mathbf{x}\big)^T\Big] \\
&= \mathbb{E}_{\mathbf{C},\mathbf{y}}\Big[\mathbf{C}^\dagger\mathbf{y}\mathbf{y}^T(\mathbf{C}^\dagger)^T\Big] + \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\mathbf{C}^\dagger\mathbf{y}\mathbf{e}^T(\mathbf{C}^\dagger)^T\Big] + 2p\mathbb{E}_{\mathbf{C},\mathbf{y}}\Big[\mathbf{C}^\dagger\mathbf{y}\mathbf{x}^T\Big] \\
&\quad + \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\mathbf{C}^\dagger\mathbf{e}\mathbf{y}^T(\mathbf{C}^\dagger)^T\Big] + \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\mathbf{C}^\dagger\mathbf{e}\mathbf{e}^T(\mathbf{C}^\dagger)^T\Big] + 2p\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\mathbf{C}^\dagger\mathbf{e}\mathbf{x}^T\Big] \\
&\quad + 2p\mathbb{E}_{\mathbf{C},\mathbf{y}}\Big[\mathbf{x}\mathbf{y}^T(\mathbf{C}^\dagger)^T\Big] + 2p\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\Big[\mathbf{x}\mathbf{e}^T(\mathbf{C}^\dagger)^T\Big] + 4p^2\mathbb{E}_{\mathbf{C},\mathbf{y}}\Big[\mathbf{x}\mathbf{x}^T\Big]\,.
\end{aligned}$$

We solve all the terms individually. Since $\mathbf{x}$ is constant, several terms are straightforward. According to Equation (24),

$$\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger\mathbf{e}\mathbf{x}^T\right] = \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger\mathbf{e}\right]\mathbf{x}^T = -2p\mathbf{x}\mathbf{x}^T = \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}[\mathbf{x}\mathbf{e}^T(\mathbf{C}^\dagger)^T]\,.$$

Furthermore, as $\mathbf{y}$ is independent of $\mathbf{C}$ with $\mathbb{E}[\mathbf{y}] = 0$, we obtain

$$\mathbb{E}_{\mathbf{C},\mathbf{y}}[\mathbf{C}^\dagger\mathbf{y}\mathbf{x}^T] = \mathbb{E}_{\mathbf{C}}[\mathbf{C}^\dagger]\mathbb{E}_{\mathbf{y}}[\mathbf{y}]\,\mathbf{x}^T = 0 = \mathbb{E}_{\mathbf{C},\mathbf{y}}[\mathbf{x}\mathbf{y}^T(\mathbf{C}^\dagger)^T]\,.$$

We further note that a scaled identity matrix is equal to its respective transpose.

We solve the more in involved terms in a series of lemmas, which are deferred to Appendix B. Our approximation comes from the use of Lemma 13 with neglecting the additional factor:

– $\mathbb{E}_{\mathbf{C},\mathbf{y}}\Big[\mathbf{C}^\dagger\mathbf{y}\mathbf{y}^T(\mathbf{C}^\dagger)^T\Big] \approx \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n$ according to Lemma 17, Appendix B.

– $\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^{\dagger}\mathbf{e}\mathbf{e}^T(\mathbf{C}^{\dagger})^T\right] \approx 4\left(p\beta^2\frac{n}{m\tau}\mathbf{I}_n + p^2\mathbf{x}\mathbf{x}^T\right)$ according to Lemma 18, Appendix B.
– $\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^{\dagger}\mathbf{y}\mathbf{e}^T(\mathbf{C}^{\dagger})^T\right] \approx \frac{p}{m}\left(\frac{-\beta^2 n}{\tau} + \|\mathbf{x}\|_2^2\right)\mathbf{I}_n$ according to Lemma 19, Appendix B.

Substituting these terms yields

$$\begin{aligned}
\mathrm{Cov}[\hat{\mathbf{x}}] &\approx \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n + 2\frac{p}{m}\left(\frac{-\beta^2 n}{\tau} + \|\mathbf{x}\|_2^2\right)\mathbf{I}_n + 2\cdot 0 \\
&\quad + 4\left(p\beta^2\frac{n}{m\tau}\mathbf{I}_n + p^2\mathbf{x}\mathbf{x}^T\right) - 4p(2p\mathbf{x}\mathbf{x}^T) + 4p^2\mathbf{x}\mathbf{x}^T \\
&= \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n + \frac{p}{m}\left(\frac{-2\beta^2 n}{\tau} + 2\|\mathbf{x}\|_2^2\right)\mathbf{I}_n + 4p\beta^2\frac{n}{m\tau}\mathbf{I}_n \\
&\quad + 4p^2\mathbf{x}\mathbf{x}^T - 8p^2\mathbf{x}\mathbf{x}^T + 4p^2\mathbf{x}\mathbf{x}^T \\
&= \frac{1}{m}\left(\frac{n\sigma_y^2}{\tau} + 2p\left(\frac{n\beta^2}{\tau} + \|\mathbf{x}\|_2^2\right)\right)\mathbf{I}_n .
\end{aligned}$$

□

As a consequence of Estimate 14, the covariance matrix tends to zero as m increases. This remains true even after rescaling by r, which multiplies the covariance according to

$$\mathrm{Cov}[r\hat{\mathbf{x}}] = r^2\mathrm{Cov}[\hat{\mathbf{x}}] . \tag{30}$$

Since the rescaled estimator $r\hat{\mathbf{x}}$ is unbiased ($\mathbb{E}[r\hat{\mathbf{x}}] = \mathbf{x}$) and its covariance vanishes, we conclude that the rescaled Least Squares estimator converges to the true subkey $\mathbf{x}$.

7.3. *Sample Complexity of Least Squares Estimator*

We proceed by providing an asymptotic analysis of the sample complexity of the least squares estimator, i.e., the number of samples required to successfully recover the secret subkey in Algorithm 3. Recall that we already established that the rescaled least squares estimator $r\hat{\mathbf{x}}$ is unbiased and quantified its covariance. Building on these results, we now quantify the convergence rate with help of the Lindeberg-Feller central limit theorem, which implies that the estimation error follows a Gaussian distribution. Consequently, we employ a Gaussian tail bound to derive the sample complexity required for successful subkey recovery for a specific success probability.

Estimate 15 (Sample complexity). *Let* $\mathbf{C}, \mathbf{z}$ *be defined as in Definition 10 for an error rate* $p \in [0, 0.5)$. *For any* $\delta \in [0, 1]$, *the rescaled least squares estimator* $r\hat{\mathbf{x}} = \frac{1}{(1-2p)}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z}$ *recovers the secret* $\mathbf{x}$ *after rounding with a probability of at least* $(1-\delta)$ *given*

$$m \geq 8\ln\left(\frac{2n}{\delta}\right)\frac{1}{(1-2p)^2}\left(\frac{n}{\tau}\left(\frac{\beta(\beta+1)}{3} + 2p\beta^2\right) + 2p\|\mathbf{x}\|_2^2\right) \tag{31}$$

informative relations.

Justification. For completeness, we first require that $\mathbf{C}^T\mathbf{C}$ is invertible for the least squares estimator to exist. According to Lemma 11, this already occurs with probability $1 - ne^{-\frac{m}{2n}}$ whenever $m \geq 2n\ln(n)$. This bound on m is small compared to Equation (31) and therefore implicitly fulfilled.

We proceed to bound the estimation error, which describes how far the estimate is from the true value.

The rescaled least squares estimator with $r = \frac{1}{1-2p}$ is unbiased according to Equation (25), and we have

$$\begin{aligned} r\hat{\mathbf{x}} - \mathbf{x} &= \frac{1}{1-2p}\hat{\mathbf{x}} - \mathbf{x} \\ &= \frac{1}{1-2p}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{z} - \mathbf{x} \\ &= \frac{1}{1-2p}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T(\mathbf{C}\mathbf{x} + \mathbf{y} + \mathbf{e}) - \mathbf{x} \\ &= \frac{1}{1-2p}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T(\mathbf{y} + \mathbf{e}) + \frac{1}{1-2p}\mathbf{x} - \mathbf{x} \\ &= \frac{1}{1-2p}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T(\mathbf{y} + \mathbf{e}) + \frac{2p\mathbf{x}}{1-2p}. \end{aligned}$$

We investigate each coefficient $j \in \{1, \ldots, n\}$ separately. Each component of the estimation error $(r\hat{\mathbf{x}} - \mathbf{x})_j$ can be expressed as a sum of random variables $Y_i^{(m)}$:

$$(r\hat{\mathbf{x}} - \mathbf{x})_j = \frac{1}{1-2p}\sum_{i=1}^{m}\Bigg(\underbrace{\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\right)_{ji}(y_i + e_i) + \frac{2px_j}{m}}_{=:Y_i^{(m)}}\Bigg).$$

First, observe that the terms $c_{ij}(y_i + e_i)$ are independent for every sample i (Definition 10). In addition, for this estimate we build on Lemma 13 (and Equation (29)) that shows $(\mathbf{C}^T\mathbf{C})^{-1} \to \frac{n}{m\tau}\mathbf{I}_n$, and thus asymptotically guarantees independence of the random variables $Y_i^{(m)}$.

We know from Theorem 12 and Equation (25) that $\mathbb{E}[Y_i^{(m)}] = 0$, and from Estimate 14 that $\mathrm{Var}[Y_i^{(m)}] = \sigma_i^2 < \infty$. More precisely, in Estimate 14 we already established that $s_m^2 = \sum_{i=1}^m \sigma_i^2 = \frac{1}{m}\sigma_{\hat{\mathbf{x}}}^2$ (where $\sigma_{\hat{\mathbf{x}}}^2$ is the scaling factor of the diagonal covariance established in Estimate 14, not involving $\frac{1}{m}$).

Following [6, Theorem 27.3 and Example 27.4], it suffices to show that Lyapounov's condition is satisfied which in turn implies the Lindeberg-Feller central limit theorem [6, Theorem 27.2].

We proceed to show that Lyapounov's condition (for its parameter $\delta = 1$) holds:

$$\lim_{m\to\infty} \frac{1}{s_m^3} \sum_{i=1}^{m} \mathbb{E}\left[|Y_i^{(m)}|^3\right] = \lim_{m\to\infty} \frac{m^{3/2}}{\sigma_{\hat{\mathbf{x}}}^3} \sum_{i=1}^{m} \mathbb{E}\left[\left|\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\right)_{ji}(y_i + e_i) + \frac{2p}{m}x_j\right|^3\right].$$

We use of the asymptotic argument from Lemma 13 and Equation (29) again: $(\mathbf{C}^T\mathbf{C})^{-1}$ is arbitrarily close to $\frac{n}{m\tau}\mathbf{I}_n$ for m large enough. Furthermore, entries in $\mathbf{C}$ as well as $(y_i + e_i)$ and x_j are bounded, which implies for some $K > 0$ and m large enough $\left|\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\right)_{ji}(y_i + e_i) + \frac{2p}{m}x_j\right| \leq \frac{K}{m}$. It follows that

$$\leq \lim_{m\to\infty} \frac{m^{3/2}}{\sigma_{\hat{\mathbf{x}}}^3} \sum_{i=1}^{m} \frac{K^3}{m^3} = \lim_{m\to\infty} \frac{m^{3/2}}{\sigma_{\hat{\mathbf{x}}}^3} \frac{K^3}{m^2} = \lim_{m\to\infty} \frac{K^3}{\sigma_{\hat{\mathbf{x}}}^3} \frac{1}{\sqrt{m}} = 0\,.$$

Therefore, the Lindeberg-Feller version of the central limit theorem applies, and the estimation error per coefficient converges to a normal distribution:

$$(r\hat{\mathbf{x}} - \mathbf{x})_j \sim \mathcal{N}\Bigg(0, \underbrace{\frac{1}{m}\frac{1}{(1-2p)^2}}_{\text{effect of rescaling}} \underbrace{\left(\frac{n\sigma_y^2}{\tau} + 2p\frac{n\beta^2}{\tau} + 2p\|\mathbf{x}\|_2^2\right)}_{\sigma_{\hat{\mathbf{x}}}^2}\Bigg).$$

Note that now the original variance $\sigma_{\hat{\mathbf{x}}}^2$ (derived in Estimate 14) is amplified with the rescaling $r = \frac{1}{1-2p}$ according to Equation (30).

We are now ready to apply a Gaussian tail bound for each component

$$\Pr[|r\hat{x}_j - x_j| > \delta_1] \leq 2\exp\left(-\frac{\delta_1^2 m(1-2p)^2}{2\sigma_{\hat{\mathbf{x}}}^2}\right),$$

and utilize the union bound to cover all n dimensions

$$\Pr[\exists i : |r\hat{x}_j - x_j| > \delta_1] \leq 2n\exp\left(-\frac{\delta_1^2 m(1-2p)^2}{2\sigma_{\hat{\mathbf{x}}}^2}\right).$$

In order to round $r\hat{\mathbf{x}}$ successfully to $\mathbf{x}$, we need to establish that *no* coordinate of $r\hat{\mathbf{x}}$ 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 at least $(1 - \delta)$, we can rearrange for the sample complexity by

Table 3. Average run time for ordinary least squares regression, experimentally required number m of informative relations in Algorithm 4 (compare with Fig. 8), our estimate Eq. (31) from Estimate 15, and the estimate Eq. (32) from [36] in the error-free setting.

Scheme	Regression [seconds]	exp. m [million]	Estimate 15 [million]	[36] [million]
ML-DSA-44	4	0.50	0.75	11
ML-DSA-87	7	0.75	1.15	17
ML-DSA-65	40	2.4	3.7	54
Ring-LWE-1024	140	2.6	3.6	52

bounding the failure probability $\delta \in [0, 1]$:

$$\begin{aligned}\Pr\left[\|r\hat{x}_j - x_j\|_\infty > \frac{1}{2}\right] &\leq 2n \exp\left(-\frac{m(1-2p)^2}{8\sigma_{\hat{\mathbf{x}}}^2}\right) \leq \delta \\ \implies m &\geq \frac{8\sigma_{\hat{\mathbf{x}}}^2}{(1-2p)^2} \ln\left(\frac{2n}{\delta}\right) \\ &= \frac{8(n\sigma_y^2 + 2p(n\beta^2 + \tau\|\mathbf{x}\|_2^2))}{\tau(1-2p)^2} \ln\left(\frac{2n}{\delta}\right).\end{aligned}$$

Plugging in $\sigma_y^2 = \frac{\beta(\beta+1)}{3}$ and simplifying gives Equation (31). □

7.4. *Tightness of our Bound and Comparison with [36]*

In Tables 3 and 4, we compare our bound for m, the required number of informative relations, from Equation (31) in Estimate 15 to our experimental data from Sect. 8, and to the bound in [36]. The bound in [36] was derived from the work in [5]. In the noisy case $p > 0$, Liu et al. obtain an additional scaling factor of $\frac{5-3p}{2(1-2p)^2}$. For a fair comparison, we instantiate both bounds with success probability $\frac{1}{2}$, which gives for [36] the formula

$$m \geq 128 \ln(2n) \frac{2+3p}{2(1-2p)^2} \frac{\beta(\beta+1)n}{3\tau}. \tag{32}$$

From Table 3, we see that our bound from Estimate 15 accurately matches the experimentally required amount of informative relations from column exp. m, and that we significantly improve over the estimate from [36]. Our improvement comes from tailoring our analysis in Sect. 6 to leaky LWE signatures for ML-DSA, whilst the analysis in [36] relies on general sub-Gaussian tail bounds.

From Table 4, we observe that in the presence of noise, our bound is also more accurate than that of [36]. Further, the ratio of experimentally observed equations to Equation (31) does not show any trend, from which we gather that we captured all contributing factors. While it still holds as a bound, this is not the case for Equation (32).

Table 4. Experimentally required number m of informative relations in Algorithm 4 (compare with Fig. 11), our estimate Eq. (31) from Estimate 15, and the estimate Eq. (32) from [36] for ML-DSA-44 parameters with error rate p.

p	exp. m [million]	Estimate 15 [million]	[36] [million]
0	0.50	0.75	11
0.1	1.4	1.9	19
0.2	3.1	4.6	39
0.3	9.0	13	98
0.4	47	64	430

7.5. *Estimating the Scale for an Unknown Error Rate*

We already established in Theorem 12 and Algorithm 3 that for an error rate p, the secret $\mathbf{x}$ can be recovered by scaling the estimate $\hat{\mathbf{x}}$ by a factor $r = 1/(1-2p)$, i.e., $\hat{\mathbf{x}} = r^{-1}\mathbf{x}$. In practice, however, the error rate is usually unknown, and therefore the scaling factor must be estimated.

Recall that the secret subkey is sampled coordinate-wise from some distribution $\mathcal{D}$. Let $X \sim \mathcal{D}$ be a random variable for a secret key coordinate. Then,

$$\mathrm{Var}\left(r^{-1}X\right) = r^{-2} \cdot \mathrm{Var}(X)\,.$$

Since $\mathcal{D}$, and hence $\mathrm{Var}(X)$ is known, and the coordinates of $\hat{\mathbf{x}}$ are sampled from the scaled distribution $r^{-1}\mathcal{D}$, it suffices to estimate the variance of $\hat{\mathbf{x}}$ to recover the unknown scaling factor r. This approach is introduced in [7], and we rely on their result.

Lemma 16. (Rescaling (adapted from Lemma 5 in [7])) *Let $\mathbf{x} \in \mathbb{R}^n$ be a secret subkey sampled coordinate-wise from the uniform distribution $\mathcal{U}$ over $[\pm\eta]$, and let $\hat{\mathbf{x}} = r^{-1}\mathbf{x}$ for some unknown $r \in \mathbb{R}$. Then*

$$\hat{r} = \frac{\sqrt{n\mathrm{Var}(\mathcal{U})}}{\|\hat{\mathbf{x}}\|_2} = \frac{\sqrt{\eta(\eta+1)n/3}}{\|\hat{\mathbf{x}}\|_2}$$

is an asymptotically optimal estimator of the scaling factor r.

8. Experimental Results

We implemented our Leaky-Signature-LWE attack (Algorithm 4)[2], 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 is 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-

[2]The implementation is publicly available on Github.

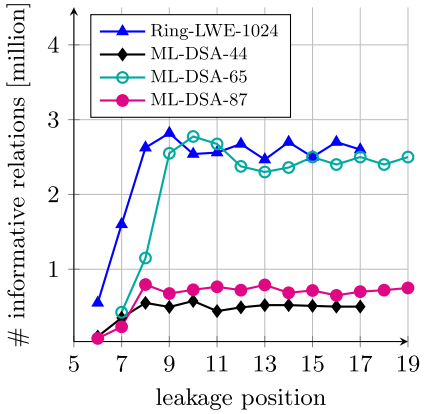

Fig. 8. Required informative relations to recover the secret key in Ring-LWE-1024 or the subkey in Dilithium in the error-free setting.

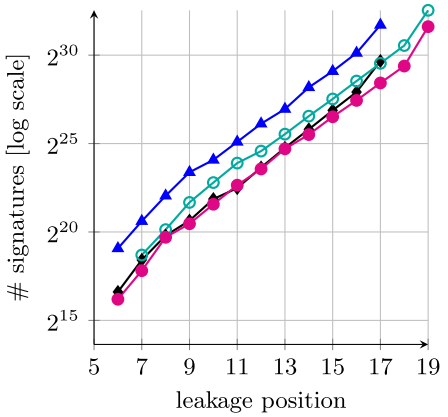

Fig. 9. Required leaky signatures to recover the secret key in Ring-LWE-1024 or the subkey in Dilithium in the error-free setting.

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 known randomness bit y_j.

Our results for the error-free setting are depicted in Figs. 8, 9 and 10, and our results for the noisy setting with an error rate $p \in [0, 0.5)$ are shown in Fig. 11.

Amount of informative relations. Figure 8 nicely illustrates the effect of filter-and-shift (Sect. 4.2), which for any leak position $j > \log_2 \beta$ reduces the ILWE error to the same uniform distribution in $[\pm\beta]$. Therefore, the same ILWE instance has to be solved, requiring roughly the same amount of informative relations for successful subkey recovery.

By Equation (8), the smaller β, the fewer relations should be required. This observation is in line with the results in Fig. 8. ML-DSA-44 with its small $\beta = 78$ (see Table 1) requires the smallest amount of 0.5 million informative relations for $j > \log_2 \beta$. For

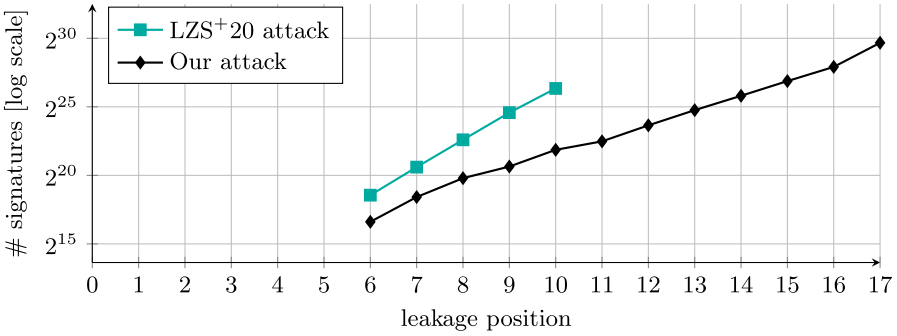

Fig. 10. Required amount of leaky signatures to recover the secret subkey in Dilithium-II (ML-DSA-44) in the error-free setting.

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 least squares 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 Fig. 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 by Equation (7) the number of *zero-knowledge* relations, that we filtered out, also roughly doubles.

Interestingly, the amount of signatures that we require is almost alike for ML-DSA-44 and ML-DSA-87, although by Fig. 8 we need significantly more informative relations for ML-DSA-87. This follows also from Equation (7) as 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 [36]. As can be seen in Fig. 10, for the most favorable leak position $j = 6$ we require roughly 100.000 leaky signatures, while [36] require roughly 4 times as many. This improvement is not due to filter-and-shift, since for $j = 6$ all samples are informative. Instead, it stems from our novel approach of working in the two's complement which halves the error size. By Equation (2), halving the error size reduces the number of samples required for least squares by a factor of 4.

In contrast to our factor 2 increase for each increment in j, [36]'s attack increases by a factor 4. This is again explained by Equation (2), as the error size in [36] doubles with every increment of j, which then leads to the factor 4 increase in the required number of signatures.

Noisy leakage. We evaluated our attack under noisy leakage for an error rate $p \in [0, 0.5)$, see Definition 7. Figure 11 shows that the secret subkey can be recovered in

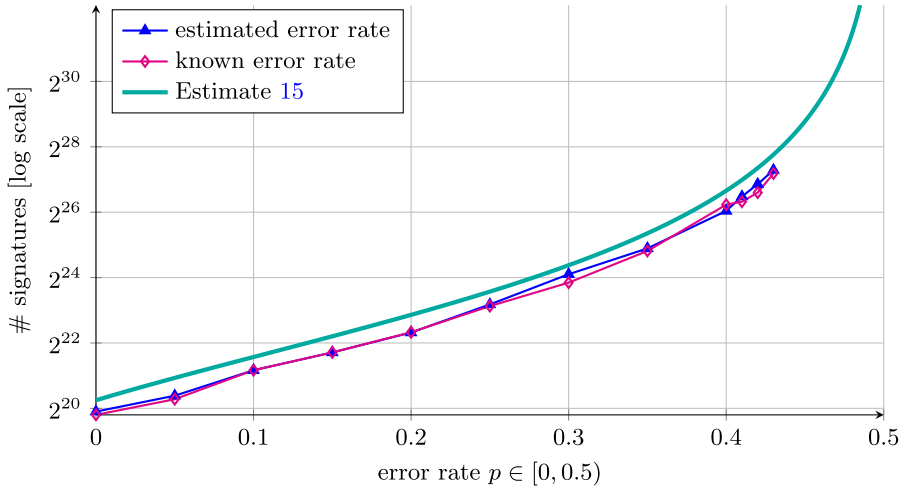

Fig. 11. Required amount of leaky signatures to recover the secret subkey in ML-DSA-44 for leakage position $j = 8$ and different error rates $p \in [0, 0.5)$.

practice for $p \leq 0.43$. This is consistent with our theoretical analysis in Sect. 6, where we established that recovery is possible for any $p < 0.5$. Remarkably, key recovery with an unknown—and hence estimated—error rate p and scaling factor r as described in Sect. 7.5 and Lemma 16 requires roughly as many signatures as key recovery with a known error rate.

We further include the theoretical bound from Estimate 15, scaled by a factor of approximately 1.65 to account for the number of signatures rather than informative relations in ML-DSA-44 at leakage position $j = 8$. We observe from Fig. 11 that our experimental results are very close to this theoretical bound.

9. Practical Considerations

In this work, we focus on the leakage of a single bit y_j at a fixed index j for all obtained signatures. This allows for a clear exposition, yet is quite restrictive in practice. We now briefly discuss how our attack generalizes to more settings.

Cross-index leakage. In the case of leakage across distinct indices $j' \geq \log_2(\beta)$, our relation extraction (Algorithm 4) can be called with a different index j for each leak. The exception is when multiple bits leak for the same coefficient in a single signature. Then, it is natural to use only the leak bit from the smallest index, since extraction from the higher-order leak bits yields the same informative relation when successful, but with a smaller extraction probability. (Recall that the probability of extracting an informative relation halves with every increase in the leak bit position.)

Multi-bit leakage. Our randomness leakage attack becomes most powerful, when a leak bit is obtained on each of the $\ell \cdot n$ signature coefficients per signature. Then, for each of the ℓ subkeys, $n = 256$ relations can be extracted per signature component

$\mathbf{z}_1 \in \mathbb{Z}^{\ell \times n}$. In this setting, the number of signatures required for key recovery would drop by a factor of 256 compared to Figs. 9, 10 and 11.

Positional noise. In practice, the exact leakage index j is not always known. Such *positional noise* can be incorporated into a new error rate p' as follows. Let $y_j^{(p)}$ be our leak bit with error rate p as defined in Definition 7, where in addition $j \in \{1, \ldots, m\}$ is a random variable for the *unknown leakage position* with known probability distribution $(p_1, \ldots, p_m)$, i.e.,

$$\Pr[j = i] = p_i.$$

Let $f : i \mapsto p_i$. In the following we choose $j^* = \arg\max_{i \in \{1,\ldots,m\}}\{f\}$. That is, we choose the most likely leak bit position with the largest probability p_{j^*}. Assuming the y_i are independent and uniformly distributed, we obtain error $1/2$ for an incorrect position guess $j^* \neq j$, which happens with probability $1 - p_{j^*}$. Thus, we obtain

$$p' = \Pr[y_j^{(p)} \neq y_{j^*}] = p_{j^*} p + (1 - p_{j^*})\frac{1}{2} = \frac{1}{2} - p_{j^*}\left(\frac{1}{2} - p\right). \tag{33}$$

Notice that $p_{j^*} = 1$ means no positional noise, in which case we obtain the usual error rate p. From Eq. (33) we also observe that p' is minimized by choosing the largest p_{j^*}.

Acknowledgements

We thank Phong Nguyen for helpful discussions. Nicolai Kraus is funded by the German Federal Ministry of Education and Research (BMBF) project PQ-CCA. Most of this work was done while Julian Nowakowski was at Ruhr University Bochum, 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.

Funding Open Access funding enabled and organized by Projekt DEAL.

A. Proof of Lemma 11

For convenience of the reader, we restate the lemma.

Lemma 11 Let $\mathbf{C}$ be an $(m \times n)$ matrix with rows independently sampled from $[\pm 1]_\tau^n$. For $m > n$, it holds that

$$\mathbb{E}_{\mathbf{C}}\left[\frac{1}{m}\mathbf{C}^T\mathbf{C}\right] = \frac{\tau}{n}\mathbf{I}_n\,. \tag{22}$$

Each entry of $\frac{1}{m}\mathbf{C}^T\mathbf{C}$ converges to its expected value 0 (off-diagonal) or $\frac{\tau}{n}$ (on-diagonal) exponentially fast in m. For $m > 2n\ln(n)$, it holds that

$$\Pr[\,\mathbf{C}^T\mathbf{C} \text{ is invertible }] > 1 - ne^{-\frac{m}{2n}}\,. \tag{23}$$

Proof. We start by computing the expectation and expand

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

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

$$\mathbb{E}\left[\frac{1}{m}\mathbf{C}^T\mathbf{C}\right] = \frac{1}{m}\mathbb{E}\left[\mathbf{C}^T\mathbf{C}\right] = \frac{1}{m}\frac{\tau m}{n}\mathbf{I}_n = \frac{\tau}{n}\mathbf{I}_n\,, \tag{34}$$

which proofs Equation (22).
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

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

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

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

Finally, 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 [50, 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.,

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

We already know from Equation (34) that

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

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

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

We now set $\delta_3 = 1$ and obtain

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

The probability that $\mathbf{C}^T\mathbf{C}$ is invertible is simply the counter event, which gives Equation (23). This concludes the proof. □

B. Additional Lemmas for Estimate 14

We present the lemmas required for the proof of Estimate 14. Throughout this section, let $\mathbf{C}^\dagger = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T$.

Lemma 17. *Let* $\mathbf{C}, \mathbf{y}$ *be defined as in Definition 10. We denote the variance of* y *by* $\sigma_y^2 = \frac{\beta(\beta+1)}{3}$. *Then*

$$\mathbb{E}_{\mathbf{C},\mathbf{y}}\left[\mathbf{C}^\dagger\mathbf{y}\mathbf{y}^T(\mathbf{C}^\dagger)^T\right] \approx \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n\,.$$

Proof. We start by computing the expectation as

$$\mathbb{E}_{\mathbf{C},\mathbf{y}}\left[\mathbf{C}^\dagger\mathbf{y}\mathbf{y}^T(\mathbf{C}^\dagger)^T\right] = \mathbb{E}_{\mathbf{C},\mathbf{y}}\left[\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\right)\left((\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\right)^T\right]$$
$$= \mathbb{E}_{\mathbf{C},\mathbf{y}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}\mathbf{y}^T\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\right].$$

As $\mathbf{y}$ is independent of $\mathbf{C}$ and $\mathbb{E}[\mathbf{y}] = 0$, its variance is $\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \sigma_y^2\mathbf{I}_m$,

$$= \sigma_y^2\mathbb{E}_{\mathbf{C}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\right]$$
$$= \sigma_y^2\mathbb{E}[(\mathbf{C}^T\mathbf{C})^{-1}]$$
$$\approx \frac{n\sigma_y^2}{m\tau}\mathbf{I}_n\,,$$

where the approximation in the last line follows from Lemma 13. □

Lemma 18. *Let* $\mathbf{C}, \mathbf{e}, \mathbf{y}, \mathbf{f}$ *be defined as in Definition 10. Then*

$$\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger\mathbf{e}\mathbf{e}^T(\mathbf{C}^\dagger)^T\right] \approx 4\left(p\beta^2\frac{n}{m\tau}\mathbf{I}_n + p^2\mathbf{x}\mathbf{x}^T\right).$$

Proof. We start by expanding the vector $\mathbf{e}$. We use $\odot$ to represent element-wise multiplication.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger \mathbf{e}\mathbf{e}^T (\mathbf{C}^\dagger)^T\right] \\
&= \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\left(-2\beta \mathbf{C}^\dagger \left(\mathbf{f} \odot \operatorname{sign}(\mathbf{z})\right)\right)\left(-2\beta \mathbf{C}^\dagger \left(\mathbf{f} \odot \operatorname{sign}(\mathbf{z})\right)\right)^T\right] \\
&= 4\beta^2 \mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger \underbrace{\begin{pmatrix} f_1 \operatorname{sign}(\langle \mathbf{c}_1, \mathbf{x}\rangle + y_1) \\ f_2 \operatorname{sign}(\langle \mathbf{c}_2, \mathbf{x}\rangle + y_2) \\ \vdots \end{pmatrix}\begin{pmatrix} f_1 \operatorname{sign}(\langle \mathbf{c}_1, \mathbf{x}\rangle + y_1) \\ f_2 \operatorname{sign}(\langle \mathbf{c}_2, \mathbf{x}\rangle + y_2) \\ \vdots \end{pmatrix}^T}_{\mathbf{u}\mathbf{u}^T} (\mathbf{C}^\dagger)^T\right].
\end{aligned} \tag{35}$$

Next, we compute the expectation of $\mathbf{u}\mathbf{u}^T$ over $\mathbf{f}$ and $\mathbf{y}$, which are independent of $\mathbf{C}$. For the diagonal items $i = j$,

$$\begin{aligned}
\mathbb{E}_{\mathbf{y},\mathbf{f}}[u_i u_i] &= \mathbb{E}_{\mathbf{y},\mathbf{f}}[f_i^2 \operatorname{sign}(\langle \mathbf{c}_i, \mathbf{x}\rangle + y_i)^2] \\
&= \Pr[f_i = 1] = p\,.
\end{aligned}$$

For the off-diagonal terms $i \neq j$:

$$\begin{aligned}
\mathbb{E}_{\mathbf{y},\mathbf{f}}[u_i u_j] &= \mathbb{E}_{\mathbf{y},\mathbf{f}}[f_i \operatorname{sign}(\langle \mathbf{c}_i, \mathbf{x}\rangle + y_i)\ f_j \operatorname{sign}(\langle \mathbf{c}_j, \mathbf{x}\rangle + y_j)] \\
&= \mathbb{E}_{\mathbf{f}}[f_i f_j]\, \mathbb{E}_{\mathbf{C},\mathbf{y}}[\operatorname{sign}(\langle \mathbf{c}_i, \mathbf{x}\rangle + y_i)\ \operatorname{sign}(\langle \mathbf{c}_j, \mathbf{x}\rangle + y_j)] \\
&= \frac{p^2}{\beta^2} \langle \mathbf{c}_i, \mathbf{x}\rangle \langle \mathbf{c}_j, \mathbf{x}\rangle\,.
\end{aligned}$$

We express the entire matrix $\mathbf{u}\mathbf{u}^T$ using $\mathbf{C}\mathbf{x}$ and $(\mathbf{C}\mathbf{x})^T = \mathbf{x}^T\mathbf{C}^T$. Further, we drop the small subtrahend for the diagonal to ease computations (as this may only increase our final estimate for the sample complexity).

$$\begin{aligned}
\mathbb{E}_{\mathbf{y},\mathbf{f}}[\mathbf{u}\mathbf{u}^T] &= p\mathbf{I}_m + \frac{p^2}{\beta^2}\left(\mathbf{C}\mathbf{x}\mathbf{x}^T\mathbf{C}^T - \operatorname{diag}\left(\mathbf{C}\mathbf{x}\mathbf{x}^T\mathbf{C}^T\right)\right) \\
&\lessapprox p\mathbf{I}_m + \frac{p^2}{\beta^2}(\mathbf{C}\mathbf{x}\mathbf{x}^T\mathbf{C}^T)\,.
\end{aligned}$$

Finally, we plug $\mathbb{E}_{\mathbf{y},\mathbf{f}}[\mathbf{u}\mathbf{u}^T]$ into Equation (35) and continue:

$$\begin{aligned}
&= 4\beta^2 \mathbb{E}_{\mathbf{C}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\left(p\mathbf{I}_m + \frac{p^2}{\beta^2}(\mathbf{C}\mathbf{x}\mathbf{x}^T\mathbf{C}^T)\right)\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\right] \\
&= 4\beta^2\left(p\mathbb{E}_{\mathbf{C}}\left[(\mathbf{C}^T\mathbf{C})^{-1}\right] + \frac{p^2}{\beta^2}\mathbf{x}\mathbf{x}^T\right).
\end{aligned}$$

Using Lemma 13, we approximate

$$\approx 4\left(p\beta^2 \frac{n}{m\tau}\mathbf{I}_n + p^2\mathbf{x}\mathbf{x}^T\right).$$

□

Lemma 19. *Let* $\mathbf{C}, \mathbf{y}, \mathbf{e}, \mathbf{f}$ *be defined as in Definition 10. Then it holds that*

$$\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^\dagger \mathbf{y}\mathbf{e}^T (\mathbf{C}^\dagger)^T\right] \approx \frac{p}{m}\left(\frac{-\beta^2 n}{\tau} + \|\mathbf{x}\|_2^2\right)\mathbf{I}_n\,.$$

Proof. As in Lemma 18, we first solve the expectation over $\mathbf{f}$ and expand $\mathbf{e}$.

$$\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^{\dagger}\mathbf{y}\mathbf{e}^{T}(\mathbf{C}^{\dagger})^{T}\right] = -2\beta p\mathbb{E}_{\mathbf{C},\mathbf{y}}\left[\mathbf{C}^{\dagger}\mathbf{y}\begin{pmatrix}\operatorname{sign}(\langle\mathbf{c}_1,\mathbf{x}\rangle+y_1)\\ \operatorname{sign}(\langle\mathbf{c}_2,\mathbf{x}\rangle+y_2)\\ \vdots\end{pmatrix}^{T}(\mathbf{C}^{\dagger})^{T}\right]. \tag{36}$$

Next, we consider the off-diagonal of the inner terms element-wise. Their expectation is zero as different samples i and j are independent, which in turn implies $\mathbb{E}_{\mathbf{C},\mathbf{y}}[y_i\operatorname{sign}(z_j)] = \mathbb{E}_{\mathbf{C},\mathbf{y}}[y_i]\mathbb{E}_{\mathbf{C},\mathbf{y}}[\operatorname{sign}(z_j)] = 0$. We compute the on-diagonal element-wise as well for any y and $\mathbf{c}$. As they are independent, we first solve for y and treat $\mathbf{c}$ as a constant;

$$\begin{aligned}
\mathbb{E}_y[y\operatorname{sign}(\langle\mathbf{c},\mathbf{x}\rangle+y)] &= \mathbb{E}_y[-y \mid \langle\mathbf{c},\mathbf{x}\rangle+y<0]\Pr[\langle\mathbf{c},\mathbf{x}\rangle+y<0]\\
&\quad+\mathbb{E}_y[y \mid \langle\mathbf{c},\mathbf{x}\rangle+y\geq 0]\Pr[\langle\mathbf{c},\mathbf{x}\rangle+y\geq 0]\\
&= \mathbb{E}_y[-y \mid y<-\langle\mathbf{c},\mathbf{x}\rangle]\Pr[y<-\langle\mathbf{c},\mathbf{x}\rangle]\\
&\quad+\mathbb{E}_y[y \mid y\geq-\langle\mathbf{c},\mathbf{x}\rangle]\Pr[y\geq-\langle\mathbf{c},\mathbf{x}\rangle]\\
&= -\frac{1}{2}\int_{-\beta}^{-\langle\mathbf{c},\mathbf{x}\rangle}\frac{1}{2\beta}y\,dy+\frac{1}{2}\int_{-\langle\mathbf{c},\mathbf{x}\rangle}^{\beta}\frac{1}{2\beta}y\,dy\\
&= \frac{1}{4\beta}\left(y^2|_{-\langle\mathbf{c},\mathbf{x}\rangle}^{\beta}-y^2|_{-\beta}^{-\langle\mathbf{c},\mathbf{x}\rangle}\right)\\
&= \frac{1}{4\beta}\left(\beta^2-\langle\mathbf{c},\mathbf{x}\rangle^2-\left(\langle\mathbf{c},\mathbf{x}\rangle^2-\beta^2\right)\right)\\
&= \frac{1}{2\beta}(\beta^2-\langle\mathbf{c},\mathbf{x}\rangle^2).
\end{aligned}$$

Next, we compute $\mathbb{E}_{\mathbf{c}}[\langle\mathbf{c},\mathbf{x}\rangle^2] = \mathbb{E}_{\mathbf{c}}[\sum_k^n c_k^2x_i^2] = \frac{\tau}{n}\sum_k^n x_i^2 = \frac{\tau}{n}\|\mathbf{x}\|_2^2$. We continue from Equation (36). To solve the expectation over $\mathbf{C}$, we use lemma 13 again. We write $\langle\mathbf{C}\mathbf{x}\rangle^2 = \mathbf{x}^T\mathbf{C}^T\mathbf{C}\mathbf{x}$ and obtain

$$\begin{aligned}
\mathbb{E}_{\mathbf{C},\mathbf{y},\mathbf{f}}\left[\mathbf{C}^{\dagger}\mathbf{y}\mathbf{e}^{T}(\mathbf{C}^{\dagger})^{T}\right] &= -2\beta p\mathbb{E}_{\mathbf{C}}\left[\mathbf{C}^{\dagger}\operatorname{diag}\left(\frac{\beta^2-\mathbf{x}^T\mathbf{C}^T\mathbf{C}\mathbf{x}}{2\beta}\right)(\mathbf{C}^{\dagger})^{T}\right]\\
&\approx -2\beta p\frac{\beta^2-\frac{\tau}{n}\|\mathbf{x}\|_2^2}{2\beta}\frac{n}{m\tau}\mathbf{I}_n\\
&= \frac{p}{m}\left(\frac{-\beta^2 n}{\tau}+\|\mathbf{x}\|_2^2\right)\mathbf{I}_n.
\end{aligned}$$

□

References

[1] M.R. Albrecht, D.J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. Von Maurich, R. Misoczki, R. Niederhagen, et al. Classic mceliece: conservative code-based cryptography. (2022)

[2] M.R. Albrecht, N. 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*. LNCS, vol. 12696 (Springer, Cham, 2021), pp. 528–558

[3] A. Akavia, Solving hidden number problem with one bit oracle and advice, In Shai Halevi, editor, *CRYPTO 2009*. LNCS, vol. 5677 (Springer, Berlin, 2009), pp 337–354

[4] D.F. Aranha, F.R. Novaes, A. Takahashi, M. Tibouchi, Y. 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*. (ACM Press, 2020), pp. 225–242

[5] J. Bootle, C. Delaplace, T. Espitau, P.A. Fouque, M. Tibouchi, LWE without modular reduction and improved side-channel attacks against BLISS, In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*. LNCS, vol. 11272 (Springer, Cham, 2018), pp. 494–524
[6] P. Billingsley, *Probability and Measure*. [u.a, 3. ed. edition] (Wiley, New York, 1995)
[7] M. Brinkmann, N. Kraus, A. May. Halfspace learning for lattice signature key recovery from signs. Cryptology ePrint Archive, Paper 2026/1366, 2026. To appear at Crypto (2026)
[8] D. Boneh, R. Venkatesan, Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes, In Neal Koblitz, editor, *CRYPTO'96*. LNCS, vol. 1109 (Springer, Berlin, 1996), pp. 129–142
[9] N. Courtois, M. Finiasz, N. Sendrier, How to achieve a McEliece-based digital signature scheme, In Colin Boyd, editor, *ASIACRYPT 2001*. LNCS, vol. 2248 (Springer, Berlin, 2001), pp. 157–174
[10] S. Checkoway, R. Niederhagen, A. Everspaugh, M. Green, T. Lange, T. Ristenpart, D.J. Bernstein, J. Maskiewicz, H. Shacham, M. Fredrikson. On the practical exploitability of dual EC in TLS implementations, In Kevin Fu and Jaeyeon Jung, editors, *USENIX Security 2014*. (USENIX Association, 2014), pp. 319–335
[11] Cve-2024-31497. https://nvd.nist.gov/vuln/detail/CVE-2024-31497. Accessed: 30.04.2024.
[12] D. Dachman-Soled, L. Ducas, H. Gong, M. Rossi. LWE with side information: attacks and concrete security estimation, in Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*. LNCS, vol. 12171 (Springer, Cham, 2020), pp. 329–358
[13] L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky. Lattice signatures and bimodal gaussians, In *Annual Cryptology Conference*. (Springer, 2013) pp. 40–56
[14] J. Devevey, P. Fallahpour, A. Passelègue, D. Stehlé. A detailed analysis of Fiat-Shamir with aborts, In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*. LNCS, vol. 14085 (Springer, Cham, 2023), pp. 327–357
[15] D. Dachman-Soled, H. Gong, T. Hanson, H. Kippen, Revisiting security estimation for LWE with hints from a geometric perspective, In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*. LNCS, vol. 14085 (Springer, Cham, 2023) pp. 748–781
[16] E. De Mulder, M. Hutter, M. E. Marson, Pete 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*. LNCS, vol. 8086 (Springer, Berlin, 2013), pp. 435–452
[17] T. Espitau, P.A. Fouque, Benoît Gérard, M. 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*. (ACM Press, 2017) pp. 1857–1874
[18] J.C. Faugere, V. Gauthier-Umana, A. Otmani, L. Perret, J. P. Tillich, A distinguisher for high-rate mceliece cryptosystems. *IEEE Trans. Inf. Theory*, **59**(10), 6830–6844 (2013)
[19] M. Finiasz, N. Sendrier. Security bounds for the design of code-based cryptosystems, In Mitsuru Matsui, editor, *ASIACRYPT 2009*. LNCS, vol. 5912 (Springer, Berling, 2009), pp. 88–105
[20] S. D. Galbraith, Space-efficient variants of cryptosystems based on learning with errors. https://www.math.auckland.ac.nz/~sgal018/compact-LWE.pdf, (2013)
[21] O. Goldreich, S. Goldwasser, S. Halevi, Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*. LNCS, vol. 1294. (Springer, Berlin, 1997), pp. 112–131
[22] C. Gentry, J. Jonsson, J. Stern, M. Szydlo. Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001, In Colin Boyd, editor, *ASIACRYPT 2001*. LNCS, vol. 2248. (Springer, Berlin, 2001) pp. 1–20
[23] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*. (ACM Press, 2008), pp. 197–206
[24] C. Gentry, M. Szydlo. Cryptanalysis of the revised NTRU signature scheme, In Lars R. Knudsen, editor, *EUROCRYPT 2002*. LNCS, vol. 2332 (Springer, Berlin, 2002), pp. 299–320
[25] N. A Howgrave-Graham, N. P. Smart, Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, **23**, 283–290 (2001)
[26] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, William Whyte, NTRUSIGN: Digital signatures using the NTRU lattice, In Marc Joye, editor, *CT-RSA 2003*. LNCS, vol. 2612 (Springer, Berlin, 2003) pp. 122–140

[27] G. Herold, A. May, LP solutions of vectorial integer subset sums — cryptanalysis of Galbraith's binary matrix LWE, In Serge Fehr, editor, *PKC 2017, Part I*. LNCS, vol. 10174 (Springer, Berlin, 2017), pp. 3–15
[28] J. Hoffstein, J. Pipher, J.H. Silverman, NTRU: a ring-based public key cryptosystem, In *Third Algorithmic Number Theory Symposium (ANTS)*. LNCS, vol. 1423 (Springer, 1998), pp. 267–288
[29] J. Hoffstein, J. Pipher, J.H. Silverman, NSS: an NTRU lattice-based signature scheme, In Birgit Pfitzmann, editor, *EUROCRYPT 2001*. LNCS, vol. 2045 (Springer, Berlin, 2001), pp. 211–228
[30] M. Hlavác, T. Rosa, Extended hidden number problem and its cryptanalytic applications, In Eli Biham and Amr M. Youssef, editors, *SAC 2006*. LNCS, vol. 4356 (Springer, Berlin, 2007), pp. 114–133
[31] N. Heninger, K. 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*. LNCS, vol. 13940 (Springer, Cham, 2023) pp. 147–176
[32] A. Hülsing, J. Rijneveld, J.M. Schanck, P. Schwabe, Ntru-hrss-kem-submission to the nist post-quantum cryptography project. (2017)
[33] V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, S. Bai, Crystals-dilithium: Technical report, national institute of standards and technology. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, (2022). Accessed: 2024-02-26
[34] V. Lyubashevsky. Fiat-Shamir with aborts: applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*. LNCS, vol. 5912 (Springer, Berlin, 2009), pp. 598–616
[35] V. Lyubashevsky. Basic lattice cryptography: the concepts behind kyber (ML-KEM) and dilithium (ML-DSA). Cryptology ePrint Archive, Report 2024/1287, (2024)
[36] Y. Liu, Y. Zhou, S. Sun, T. Wang, R. Zhang, J. Ming. On the security of lattice-based fiat-shamir signatures in the presence of randomness leakage. *IEEE Trans. Inf. Forensic Secur.*, **16**, 1868–1879 (2020)
[37] R. Merget, M. Brinkmann, N. Aviram, J. Somorovsky, J. Mittmann, J. Schwenk, Raccoon attack: finding and exploiting most-significant-bit-oracles in TLS-DH(E), In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*. (USENIX Association, 2021), pp. 213–230
[38] R.J McEliece, A public-key cryptosystem based on algebraic. *Coding Thv*, **4244**, 114–116 (1978)
[39] A. May, J. Nowakowski, Too many hints - when LLL breaks LWE, In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part IV*. LNCS, vol. 14441 (Springer, Singapore, 2023), pp. 106–137
[40] National Institute of Standards and Technology. Module-lattice-based digital signature standard (2024)
[41] P.Q. Nguyen, O. Regev, Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures. *J. Cryptol.* **22**(2), 139–160 (2009)
[42] Nguyen, Shparlinski, The insecurity of the digital signature algorithm with partially known nonces. *J. Cryptol.* **15**, 151–176 (2002)
[43] P.Q Nguyen, I.E. Shparlinski, The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Des. Codes Cryptogr.* **30**, 201–217 (2003)
[44] P.A. Oliveira, Andersson Calle Viera, B. Cogliati, L. Goubin, Uncompressing dilithium's public key. *IACR* Cryptol. ePrint Arch., pp. 1373, (2024)
[45] Z. Qiao, Y. Liu, Y. Zhou, J. Ming, C. Jin, H. Li, Practical public template attacks on crystals-dilithium with randomness leakages. *IEEE Trans. Inf. Forensics Secur.*, **18**, 1–14 (2022)
[46] K. Ryan, Return of the hidden number problem. *IACR* TCHES. **2019**(1), 146–168 (2018)
[47] C. P. Schnorr, A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.* **53**(2-3), 201–224, (1987)
[48] C.P. Schnorr, Efficient identification and signatures for smart cards, in Gilles Brassard, editor, *CRYPTO'89*. LNCS, vol. 435 (Springer, New York, 1990), pp. 239–252
[49] G.W. Stewart, *Matrix algorithms: volume 1: basic decompositions*. SIAM, (1998)
[50] J.A. Tropp, User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, **12**, 389–434 (2012)
[51] J. Xu, S. Sarkar, H. Wang, L. Hu. Improving bounds on elliptic curve hidden number problem for ECDH key exchange, In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*. LNCS, vol. 13793 (Springer, Cham, 2022), pp. 771–799