Cryptanalysis of EagleSign

Ludo N. Pulles[1] and Mehdi Tibouchi[2]

[1] CWI Amsterdam
ludo.pulles@cwi.nl
[2] NTT Social Informatics Laboratories
mehdi.tibouchi@ntt.com

Abstract. EagleSign is one of the 40 "Round 1 Additional Signatures" that is accepted for consideration in the supplementary round of the Post-Quantum Cryptography standardization process, organized by NIST. Its design is based on structured lattices, and it boasts greater simplicity and performance compared to the two lattice signatures already selected for standardization: Falcon and Dilithium.

In this paper, we show that those claimed advantages come at the cost of security. More precisely, we show that the distribution of EagleSign signatures leaks information about the private key, to the point that only a few hundred signatures on arbitrary known messages suffice for a full key recovery, for all proposed parameters.

A related vulnerability also affects EagleSign-V2, a subsequent version of the scheme specifically designed to thwart the initial attack. Although a larger number of signatures is required for key recovery, the idea of the attack remains largely similar. Both schemes come with proofs of security that we show are flawed.

Keywords: EagleSign · Lattice-based signatures · Cryptanalysis · Fiat–Shamir with aborts

1 Introduction

Since the vast majority of currently deployed cryptographic schemes nowadays rely on the hardness of factoring or discrete logarithms, the entire foundation of modern secure communications would be at risk if large-scale quantum computers capable of running Shor's algorithm were to come about. As that perspective appears less remote than it did in the past, the cryptographic community has put great efforts into the transition to post-quantum schemes, based on assumptions not known to be broken by quantum computers.

These efforts have been spurred in particular by NIST's post-quantum cryptography (PQC) standardization process [NIST17], as well as competitions launched by other standardization bodies [CPQC, KPQC]. NIST's process, in particular, started in 2016, and reached an important milestone in 2022 when a first batch of 4 algorithms (three signatures schemes and one KEM) were selected for standardization. A handful of other KEM schemes remain under consideration for future standardization, and NIST also called for the submission of additional signature schemes to be examined in a similar process as the original one [NIST23]. This additional process began in 2023 and is currently ongoing.

One stated motivation for NIST's call for additional signatures is that among the three signature schemes already selected for standardization, two are based on structured lattices (Dilithium [LDK+22] and FALCON [PFH+22]), and the last scheme, SPHINCS+ [HBD+22], has somewhat limited applicability due to the size and performance characteristics of hash-based signatures. NIST therefore encouraged the submission of additional signatures relying on hardness assumptions not related to structured lattices. Nevertheless, several of the submissions are indeed based on structured lattices.

Fiat–Shamir lattice-based signatures and the EagleSign signature scheme. One of those lattice-based submissions to NIST's call for additional signatures is EagleSign [SHDS23a]. It can be informally described as a Fiat–Shamir type signature scheme, like Dilithium, except that EagleSign does not involve rejection sampling. Moreover, it is in contrast with FALCON, which is a hash-and-sign type signature scheme.

Fiat–Shamir type signatures, which are ubiquitous in the discrete logarithm setting (epitomized by Schnorr signatures and variants like DSA and ECDSA), were extended to the lattice setting in large part thanks to Lyubashevsky's "Fiat–Shamir with aborts" framework [Lyu09]. Lyubashevsky's technique addresses the issue that, contrary to what happens in a finite group, one cannot sample a uniformly random element of a lattice or its ambient space, which makes it difficult for a lattice-based sigma protocol to satisfy the (honest verifier) zero-knowledge property. This is fixed in Lyubashevsky's framework by throwing away transcripts that would leak statistical information about the secret, which translates in the corresponding signature scheme to a rejection sampling step in the signature generation algorithm.

This rejection sampling step is sometimes regarded as a source of complexity, reduced efficiency and possible side-channel attack vulnerability for implementations of Fiat–Shamir lattice-based signatures [EFGT17, BBE+19, dPRS23]. Accordingly, there have been attempts to avoid it entirely. This can be achieved for example using noise flooding, as done in the Raccoon signature scheme [dEK+23] and similar constructions. This requires a superpolynomial modulus, however, and hence somewhat larger parameters and qualitatively stronger hardness assumptions.

The EagleSign signature scheme also proposes to eliminate aborts and rejection sampling, while at the same time offering more compact parameters compared to Dilithium. Although this does not, to our knowledge, contradict any known impossibility result, this is a priori surprising, and casts doubt on the security of the scheme, even though it claims to come with a proof of security.

Our contributions. In this paper, we show that such doubts are warranted, and that EagleSign is indeed insecure. Signatures tend to leak partial information on the private key, leading to a full private key recovery attack from sufficiently many valid signatures. We implement the attack, and show that a few hundred signatures on average are enough for key recovery for both proposed parameter sets. Accordingly, we point out a flaw in the security proof.

This attack was previously announced on the NIST PQC Forum [Tib23, Pul23], and acknowledged by the submitters. In response, they revised their specification with a new version of the EagleSign scheme, called *EagleSign-V2*, this time featuring rejection sampling. Unfortunately, the newly introduced rejection sampling fails to achieve the honest verifier zero knowledge property, and the various changes do not succeed in preventing a similar attack strategy. These findings were again announced on the PQC Forum with an accompanying implementation, and subsequently acknowledged by the authors [Hou23].

2 Preliminaries

2.1 Notation

Vectors and matrices. Vectors are boldfaced and, unless otherwise stated, column vectors. Matrices are boldfaced and capitalized. The transpose of a vector $\mathbf{v}$ is denoted by $\mathbf{v}^\mathsf{T}$. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$.

We define a partial order relation $\prec$ on $n \times n$ real symmetric matrices by writing $\mathbf{\Sigma}_1 \prec \mathbf{\Sigma}_2$ when $\mathbf{\Sigma}_2 - \mathbf{\Sigma}_1$ is positive semidefinite, i.e., $\forall \mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\mathsf{T}(\mathbf{\Sigma}_2 - \mathbf{\Sigma}_1)\mathbf{x} \geq 0$. We have, in particular, $\mathbf{\Sigma} \prec \alpha \mathbf{I}_n$ if and only if all the eigenvalues of $\mathbf{\Sigma}$ are at most α.

Probabilities. A random variable X following a distribution D is denoted by $X \sim D$. The probability of some event E happening is denoted by $\Pr[E(X)]$, the *expectation value* of $f(X)$ is written as $\mathbb{E}[f(X)]$. The *variance* of X is $\mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$, while the *covariance matrix* of random variables $\mathbf{X} = (X_1, \ldots, X_n)^\mathsf{T}$ is given by

$$\mathbf{\Sigma}_\mathbf{X} = \mathbb{E}\left[(\mathbf{X} - \mathbb{E}[\mathbf{X}]) \cdot (\mathbf{X}^\mathsf{T} - \mathbb{E}[\mathbf{X}^\mathsf{T}])\right].$$

The *uniform distribution on a set* X is denoted by $\mathcal{U}(X)$. The variance of $\mathcal{U}([-\eta, \eta] \cap \mathbb{Z})$ is given by $\eta(\eta + 1)/3$ for $\eta \in \mathbb{Z}_{\geq 0}$.

Given a symmetric positive definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, the (centered) *Gaussian (or multivariate normal distribution)* with covariance matrix $\mathbf{\Sigma}$ is a distribution with probability density function

$$\frac{1}{\sqrt{(2\pi)^n \cdot \det(\mathbf{\Sigma})}} \cdot e^{-\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{x}},$$

at $\mathbf{x}^\mathsf{T} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, and is denoted by $\mathcal{N}(0, \mathbf{\Sigma})$.

Cyclotomic rings. For n a power of two, R denotes $\mathbb{Z}[X]/(X^n+1)$, the ring of integers of the cyclotomic number field of degree n. The *coefficient embedding* of R is the additive group homomorphism

$$\begin{aligned}\mathrm{coef}\colon \quad & R \to \mathbb{Z}^n \\ f = f_0 + f_1X + \dots f_{n-1}X^{n-1} & \mapsto (f_0, f_1, \dots, f_{n-1})^\mathsf{T}.\end{aligned}$$

Given an element $f \in R$, we write f_i to mean $\mathrm{coef}(f)_i$, and frequently identify f with $\mathrm{coef}(f)$ by abuse of notation. The standard inner product on $\mathbb{R}^n$ induces an inner product and a lattice structure on R.

In addition, we denote for any $f \in R$ the ℓ_p-norm and ℓ_∞-norm as follows:

$$\|f\|_p = \left(\sum_{i=0}^{n-1} f_i^p\right)^{1/p} \qquad \text{(for all } p \geq 1), \qquad \|f\|_\infty = \max_{0\leq i<n}\{f_i\}.$$

The free R-module of rank k is denoted by $R^k = \underbrace{R \oplus R \oplus \cdots \oplus R}_{k \text{ times}}$. Elements of R^k are column vectors $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_k)^\mathsf{T}$, which can be thought of as a vector in $\mathbb{Z}^{nk}$ using the coefficient embedding. Given a vector $\mathbf{f} \in R^k$, let us write $\mathbf{f}_{i,j} = \mathrm{coef}(\mathbf{f}_i)_j$, for $1 \leq i \leq k$ and $0 \leq j < n$.

The number ring comes with a complex conjugation map $f \mapsto f^*$ given by

$$f^* = \sum_{i=0}^{n-1} f_i X^{-i} = f_0 + \sum_{i=1}^{n-1}(-f_{n-i})X^i,$$

for $f \in R$. Extended to modules, we have $\mathbf{f}^* = (\mathbf{f}_1^*, \dots \mathbf{f}_k^*)$ as a row vector, given a column vector $\mathbf{f} \in R^k$. Observe that $f \cdot f^*$ has constant term equal to $\|f\|_2^2$.

Reusing some notation from the EagleSign specification document [SHDS23a], we define

$$\begin{aligned} S_\eta &= \{f \in R \mid \|f\|_\infty \leq \eta\} && \text{(for any } \eta \in \mathbb{Z}_{\geq 1}), \\ B_\tau &= \{f \in R \mid \|f\|_\infty \leq 1, \text{ and } \|f\|_1 = \tau\} && \text{(for any } 0 \leq \tau \leq n), \\ B_\tau^0 &= \{f \in B_\tau \mid f_0 = 0\} && \text{(for any } 0 \leq \tau < n). \end{aligned}$$

Here, S_η is a set of polynomials with small coefficients, and B_τ is a set of sparse ternary polynomials of fixed weight.

2.2 The EagleSign-V1 signature scheme

First, we will consider the original version of EagleSign, specifically “EagleSign 2”, which can be found in Section 2.2 of the specification document [SHDS23a]. We will refer to this version as “EagleSign-V1”, to make the distinction clear between this version and EagleSign-V2 [Hou23]. For this particular case, the submitters supplied two implementations: one aiming at NIST’s security level 3,

Table 1. Parameter sets for EagleSign-V1 as taken from [SHDS23a, Table 2].

Claimed security level	3	5
n	1024	1024
k (length of $\mathbf{w}$)	1	1
ℓ (length of $\mathbf{z}$)	1	2
η_G	1	1
η_D	1	1
η_{y2}	64	32
τ (weight of $\mathbf{c}_i$)	38	18
t (weight of $\mathbf{y}_{1,i}$)	140	86
δ (bound on $\lVert\mathbf{z}\rVert_\infty$)	178	208
δ' (bound on $\lVert\mathbf{w}\rVert_\infty$)	242	240

and one aiming at security level 5. Here, we will only highlight some of the parts that are relevant for the attack.

The signature scheme works with the number ring $R = \mathbb{Z}[X]/(X^n + 1)$ and free R-modules, where n is a power of two. All the parameters for the two supplied parameter sets can be found in Table 1. EagleSign-V1 uses the prime modulus $q = 12289$. Note that [SHDS23a, Table 1] contains a typo: it reports $\tau = 38$ for security level 5, but the reference implementation and [SHDS23a, Table 2] use $\tau = 18$ instead, which gives the correct length bounds.

Private key. The private key consists of three matrices: $\mathbf{A} \in (R/qR)^{k\times\ell}$, $\mathbf{G} \in S_{\eta_G}^{\ell\times\ell}$, and $\mathbf{D} \in S_{\eta_D}^{k\times\ell}$.

During key generation, the matrix $\mathbf{A}$ is sampled using a SHAKE-based pseudorandom generator that takes a seed ρ as input. Thus, in fact the matrix $\mathbf{A}$ is not stored, but instead ρ is stored, and $\mathbf{A}$ is computed when needed.

The matrices $\mathbf{G}$ and $\mathbf{D}$ are generated by a function that samples small uniform polynomials using a pseudorandom generator. Additionally, key generation repeats sampling $\mathbf{G}$ until it is invertible modulo q.

Public key. The public key consists of two matrices: $\mathbf{A}$, and $\mathbf{E} \in (R/qR)^{k\times\ell}$, which is given by

$$\mathbf{E} \equiv (\mathbf{A} + \mathbf{D})\,\mathbf{G}^{-1} \pmod{qR^{k\times\ell}}. \tag{1}$$

Signature generation. A high-level idea of the algorithm to sign a message is found in Algorithm 1. Here, the functions $\mathcal{G}\colon (R/qR)^k \to \{0,1\}^{256}$ and $\mathcal{H}\colon \{0,1\}^* \times \{0,1\}^{256} \to B_\tau^\ell$ are collision-resistant hash functions based on SHAKE. Algorithm 1 omits two technical details:

Algorithm 1 EagleSign-V1-Sign$(m, (\mathbf{A}, \mathbf{G}, \mathbf{D}))$

```
Require: a message $m \in \{0,1\}^*$, and a private key $(\mathbf{A}, \mathbf{G}, \mathbf{D})$.
Ensure: a valid signature $\sigma = (r, \mathbf{z}, \mathbf{w})$ for $m$.
$\mathbf{y}_1 \leftarrow \mathcal{U}\big(B_t^\ell\big)$
$\mathbf{y}_2 \leftarrow \mathcal{U}\big(S_{\eta_{y2}}^k\big)$
$r \leftarrow \mathcal{G}\big(\mathbf{A} \cdot \mathbf{y}_1 + \mathbf{y}_2 \pmod{qR^k}\big) \qquad (\in \{0,1\}^{256})$
$\mathbf{c} \leftarrow \mathcal{H}(m, r) \qquad (\in B_\tau^\ell)$
$\mathbf{z} \leftarrow \mathbf{G} \cdot (\mathbf{y}_1 + \mathbf{c}) \pmod{qR^\ell}$
$\mathbf{w} \leftarrow \mathbf{y}_2 - \mathbf{D} \cdot (\mathbf{y}_1 + \mathbf{c}) \pmod{qR^k}$
return $(r, \mathbf{z}, \mathbf{w})$
```

- The message m is directly hashed with SHAKE256, before being fed to $\mathcal{H}$, as follows:
  $$m \leftarrow \text{SHAKE256}(\text{tr} \| m)[0:384],$$
  where $\text{tr} \in \{0,1\}^{256}$ is in fact part of the public and private key, and $\|$ denotes concatenation. One can interpret tr as a salt that is fixed per key pair.
- Line 12–16 of [SHDS23a, Alg. 2] contains an optional check whether the signature is valid. This condition holds by construction, making the check unnecessary.

Verification. A signature $(r, \mathbf{z}, \mathbf{w})$ is valid for public key $(\mathbf{A}, \mathbf{E})$ and message m, whenever $\|\mathbf{z}\|_\infty \le \delta$, $\|\mathbf{w}\|_\infty \le \delta'$, and $\mathcal{H}(m, r) = \mathcal{H}(m, r')$ all hold, where

$$r' = \mathcal{G}\big(\mathbf{E}\mathbf{z} - \mathbf{A} \cdot \mathcal{H}(m, r) + \mathbf{w} \pmod{qR^k}\big)\,.$$

In order to make sure that generated signatures are valid, the length bounds δ, δ' (found in Table 1) are chosen as follows, according to [SHDS23a, p. 13]:

$$\delta = \ell \cdot \eta_G \cdot (t + \tau), \qquad \delta' = \eta_{y2} + \ell \cdot \eta_D \cdot (t + \tau).$$

One can see that the output of Algorithm 1 is always a valid signature, even when performing no operations modulo q, by definition of δ and δ'.

Note that the verification could have been more strict by demanding $r = r'$ instead of $\mathcal{H}(m, r) = \mathcal{H}(m, r')$. The latter condition allows for a weak forgery: when an attacker has a signature $\sigma = (r, \mathbf{z}, \mathbf{w})$ for a message m, and has a hash collision $\mathcal{H}(m, r) = \mathcal{H}(m, r'')$, then $(r'', \mathbf{z}, \mathbf{w})$ is a weakly forged signature for m.

2.3 The EagleSign-V2 signature scheme

EagleSign-V2 is based on the EagleSign-V1 scheme, and was announced in [Hou23]. Like EagleSign-V1, the scheme works with the number ring $R = \mathbb{Z}[X]/(X^n + 1)$, where n is a power of two, and uses R-modules. The parameter sets can be found in Table 2. Note that the claimed security level of 5++ is not one of the security requirements defined by NIST's call [NIST23].

Table 2. Parameter sets for EagleSign-V2 as taken from [SHDS23b, Table 2].

Claimed security level	2	5	5++
n	1024	2048	1024
q	2021377	33292289	7340033
k	1	1	3
ℓ (length of $\mathbf{z}$)	1	1	2
η_f (smallness of $\mathbf{F}$)	1	1	1
γ_1 (smallness of $\mathbf{y}$)	2^{14}	2^{16}	2^{16}
γ_2 (high bits)	2^{17}	2^{20}	2^{20}
t_D (weight of $\mathbf{D}_{i,j}$)	7	13	3
t_g (weight of g)	7	14	16
τ (weight of $\mathbf{c}_i$)	16	30	16

Algorithm 2 EagleSign-V2-Sign$(m, (\mathbf{A}, g, \mathbf{D}, \mathbf{F}))$

```
Require: a message $m \in \{0,1\}^*$, and a private key $(\mathbf{A}, \mathbf{G}, \mathbf{D})$.
Ensure: a valid signature $\sigma = (r, \mathbf{z})$ for $m$.
$\mathbf{y} \leftarrow \mathcal{U}(S_{\gamma_1}^{\ell})$
$\mathbf{p} \leftarrow (\mathbf{A}\mathbf{F}^{-1} + \mathbf{D})\mathbf{y} \bmod qR^k$
$r \leftarrow \mathcal{G}(\mathsf{HighBits}(\mathbf{p}, 2\gamma_2))$        $(\in \{0,1\}^{256})$
$\mathbf{c} \leftarrow \mathcal{H}(m, r)$        $(\in B_{\tau}^{\ell})$
$\mathbf{z} \leftarrow g \cdot (\mathbf{y} + \mathbf{F}\mathbf{c}) \pmod{qR^{\ell}}$
if $\|\mathbf{z}\|_{\infty} > t_g(\gamma_1 - \ell\tau)$ then
    Restart.
$\mathbf{v} \leftarrow \mathbf{p} + \mathbf{D}\mathbf{F}\mathbf{c}$                ▷ Note: $\mathbf{v} = \mathbf{E}\mathbf{z} - \mathbf{A}\mathbf{c}$
if $\|\mathbf{v}\|_{\infty} \geq \frac{q-1}{2} - \ell^2 t_D \tau$, or $\|\mathsf{LowBits}(\mathbf{v}, 2\gamma_2)\|_{\infty} \geq \gamma_2 - \ell^2 t_D \tau$ then
    Restart.
return $(r, \mathbf{z})$
```

Private key. The private key consists of four matrices: $\mathbf{A} \in (R/qR)^{k\times\ell}$, $\mathbf{F} \in S_{\eta_F}^{\ell\times\ell}$, $\mathbf{D} \in B_{t_D}^{k\times\ell}$ and $g \in B_{t_g}$. There are two differences with EagleSign-V1, namely that the matrix $\mathbf{G}$ is now a ring element $g \in R$, and also there is a new matrix $\mathbf{F}$.

Now, g is sampled using a pseudorandom generator until it is invertible, and $\mathbf{F}$ is sampled until it is invertible modulo q.

Public key. The public key is still $(\mathbf{A}, \mathbf{E})$, where now $\mathbf{E}$ is given by,

$$\mathbf{E} = \left(\mathbf{A}\mathbf{F}^{-1} + \mathbf{D}\right) g^{-1} \pmod{qR^{k\times\ell}}.$$

Signature generation. Similarly to EagleSign-V1, the function $\mathcal{G}$ and $\mathcal{H}$ are used within signing. However now $\mathcal{G}$ receives as input some “high-order bits”, similar to Dilithium. For $w \in \mathbb{Z}$, let $\mathsf{HighBits}(w, 2\gamma_2)$ be the smallest integer

$w_1 \in \mathbb{Z}$ such that $|w - 2\gamma_2 w_1| \le \gamma_2$, and let us extend this to integer vectors $\mathbf{w}$ coefficientwise as well: $\mathsf{HighBits}(\mathbf{w}, 2\gamma_2)_i = \mathsf{HighBits}(\mathbf{w}_i, 2\gamma_2)$. Moreover, let $\mathsf{LowBits}(w, 2\gamma_2) = w - 2\gamma_2 \mathsf{HighBits}(w, 2\gamma_2)$.

A high-level overview of signing can be found in Algorithm 2. There are now three possible reasons to restart during signing. While the first is concerned about the ℓ_∞-norm of $\mathbf{z}$, the other two are there to make sure that

$$\mathsf{HighBits}(\mathbf{E}\mathbf{z} - \mathbf{A}\mathbf{c}, 2\gamma_2) = \mathsf{HighBits}(\mathbf{p}, 2\gamma_2),$$

holds, which will be needed for verifying the correctness of the signature.

Verification. A signature $(r, \mathbf{z})$ is valid for public key $(\mathbf{A}, \mathbf{E})$ and message m, whenever $\|\mathbf{z}\|_\infty \le t_g(\gamma_1 - \ell\tau)$, and $\mathcal{H}(m, r) = \mathcal{H}(m, r')$, where r' is given by

$$r' = \mathcal{G}\big(\mathsf{HighBits}\big(\mathbf{E}\mathbf{z} - \mathbf{A} \cdot \mathcal{H}(m, r) \pmod{qR^k}, 2\gamma_2\big)\big) .$$

3 A Known-Message Attack on EagleSign-V1

In this section, we describe a key recovery attack on the signature scheme of Section 2.2.

Attack strategy. The basic idea of the attack can be described as follows. A valid EagleSign-V1 signature for a message reveals rather short vectors $\mathbf{z} \in R^\ell$ and $\mathbf{c} \in B_\tau^\ell$ such that:

$$\mathbf{z} = \mathbf{G} \cdot (\mathbf{y}_1 + \mathbf{c}), \tag{2}$$

where the matrix $\mathbf{G}$ is part of the private key, and $\mathbf{y}_1$ is sampled uniformly from B_t^ℓ. The random oracle model ensures that $\mathbf{c}$ is sampled uniformly from B_τ^ℓ, and is independent of $\mathbf{y}_1$.

Although the original description of the scheme includes a reduction modulo q in the computation of $\mathbf{z}$, no reduction occurs since all variables in Eq. (2) have small ℓ_∞ norm. In fact, this is necessary to prove correctness of generated signatures! As a result, Eq. (2) above does hold over the ring R.

Now, with significant probability $\tau/2n$ (which is $\approx 1.9\%$ for the level 3 parameter set and $\approx 0.9\%$ for the level 5 parameter set), we have $\mathbf{c}_{1,0} = 1$, i.e. the constant coefficient of the first entry of $\mathbf{c}$ is 1. Since the expectation values of all other coefficients of $\mathbf{c}$ are 0, and $\mathbb{E}[\mathbf{y}_1] = 0$ holds, the conditional expectation $\mathbb{E}[\mathbf{z} \mid \mathbf{c}_{1,0} = 1]$ is simply given by:

$$\mathbb{E}[\mathbf{z} \mid \mathbf{c}_{1,0} = 1] = \mathbf{G} \cdot \big(0 + (1, 0, \ldots, 0)^\mathsf{T}\big) \tag{3}$$

which is exactly $\mathbf{G}_1$, the first column of $\mathbf{G}$. As a result, an attacker can simply collect many signatures with $\mathbf{c}_{1,0} = 1$ and average the corresponding values of $\mathbf{z}$ together to recover a close approximation of $\mathbf{G}_1$. Other columns $\mathbf{G}_i$ can be recovered similarly by considering signatures with $\mathbf{c}_{i,0} = 1$ for other indices $1 \le i \le \ell$ instead.

Note that recovering $\mathbf{G}$ is sufficient for a full private key recovery: using Eq. (1) and the public key $(\mathbf{A}, \mathbf{E})$, one can compute $\mathbf{D}$ by $\mathbf{D} = \mathbf{E} \cdot \mathbf{G} - \mathbf{A}$. Alternatively, one could also use a similar strategy and analysis on the leakage from

$$\mathbf{w} = \mathbf{y}_2 - \mathbf{D} \cdot (\mathbf{y}_1 + \mathbf{c}) \,, \tag{4}$$

by also filtering on signatures with $\mathbf{c}_{1,0} = 1$, to recover $\mathbf{D}$. However, we will focus the analysis on $\mathbf{G}$ solely, as this is sufficient.

Analysis of the attack. Now, let us determine the expected number S of signatures needed to recover $\mathbf{G}_1$ with the above strategy. Given s signatures $(\mathbf{z}^{(j)}, \mathbf{c}^{(j)})_{1 \leq j \leq s}$ all having $\mathbf{c}_{1,0}^{(j)} = 1$, the attack guesses the value of $\mathbf{G}_1$ by rounding $\bar{\mathbf{z}} = \frac{1}{s} \sum_{j=1}^{s} \mathbf{z}^{(j)}$. It follows from Eq. (2) that:

$$\begin{aligned}
\bar{\mathbf{z}} &= \frac{1}{s} \sum_{j=1}^{s} \mathbf{G} \cdot \left(\mathbf{y}_1^{(j)} + \mathbf{c}^{(j)}\right) \\
&= \mathbf{G}_1 + \frac{1}{s} \sum_{j=1}^{s} \mathbf{G} \cdot \left(\mathbf{y}_1^{(j)} + (\mathbf{c}_1^{(j)} - 1, \mathbf{c}_2^{(j)}, \ldots, \mathbf{c}_\ell^{(j)})^\mathsf{T}\right) \\
&= \mathbf{G}_1 + \mathbf{G} \cdot \left(\bar{\mathbf{y}}_1 + (\bar{\mathbf{c}}_1', \bar{\mathbf{c}}_2, \ldots, \bar{\mathbf{c}}_\ell)^\mathsf{T}\right),
\end{aligned} \tag{5}$$

where $\bar{\mathbf{y}}_1$ is the mean of the random vectors $\mathbf{y}_1^{(j)}$ for each of the signature samples (unknown to the attacker), $\bar{\mathbf{c}}_i$ is the mean of the i-th entries of the vectors $\mathbf{c}^{(j)}$, and $\bar{\mathbf{c}}_1' = \bar{\mathbf{c}}_1 - 1$ is the recentered first entry, taking into account the knowledge that $\mathbf{c}_{1,0}^{(j)} = 1$ for all j.

Each of the entries of $\mathbf{y}_1^{(j)}$ is sampled from the uniform distribution on B_t. Similarly, $\mathbf{c}_2^{(j)}, \ldots, \mathbf{c}_\ell^{(j)}$ are all i.i.d. sampled from the uniform distribution on B_τ. Since $\mathbf{c}_{1,0}^{(j)} = 1$, we can also see that $\mathbf{c}_1^{(j)} - 1$ follows the uniform distribution on $B_{\tau-1}^0$. Moreover, all of the above ring elements are mutually independent.

Since the previous distributions all have mean zero, the expectation values of $\bar{\mathbf{y}}_1$, $\bar{\mathbf{c}}_2, \ldots, \bar{\mathbf{c}}_\ell$ and even $\bar{\mathbf{c}}_1'$ are all zero. Furthermore, for any $w \geq 1$, let $\mathbf{\Sigma}_w$ denote the covariance matrix of $\mathcal{U}(B_w)$, seen as a distribution over $\mathbb{Z}^n$, and let $\mathbf{\Sigma}_w^0$ denote the covariance matrix of $\mathcal{U}\big(B_w^0\big)$. The central limit theorem ensures that, as s tends to infinity, $\bar{\mathbf{y}}_1$ behaves like the normal distribution $\mathcal{N}(0, \frac{1}{s}\mathbf{\Sigma}_t \otimes \mathbf{I}_\ell)$; $\bar{\mathbf{c}}_i$ for $i \geq 2$ behaves like $\mathcal{N}(0, \frac{1}{s}\mathbf{\Sigma}_\tau)$; and $\bar{\mathbf{c}}_1'$ behaves like $\mathcal{N}(0, \frac{1}{s}\mathbf{\Sigma}_{\tau-1}^0)$.

Lemma 1. *The covariance matrix $\mathbf{\Sigma}_w$ of $\mathcal{U}(B_w)$, seen as a distribution over $\mathbb{Z}^n$, is $\frac{w}{n}\mathbf{I}_n$, and the covariance matrix $\mathbf{\Sigma}_w^0$ of $\mathcal{U}\big(B_w^0\big)$ is* $\operatorname{diag}\left(0, \frac{w}{n-1}, \ldots, \frac{w}{n-1}\right)$.

Proof. Consider the random vector $Y = (Y_0, \ldots, Y_{n-1})^\mathsf{T} \in \mathbb{Z}^n$ uniformly drawn from B_w. Since Y has zero expectation, we have $\mathbf{\Sigma}_w = \mathbb{E}[YY^\mathsf{T}]$. Each Y_i has support $\{-1, 0, 1\}$, and since the signs of each Y_i are sampled uniformly and independently, we have $\Pr[Y_iY_j = 1] = \Pr[Y_iY_j = -1]$ for all $i \neq j$. Thus, the off-diagonal coefficients of $\mathbf{\Sigma}_w$, given by $\mathbb{E}[Y_iY_j] = \Pr[Y_iY_j = 1] - \Pr[Y_iY_j = -1]$, all vanish.

Moreover, the i-th diagonal coefficient of $\mathbf{\Sigma}_w$ is given by $\mathbb{E}[Y_i^2]$, which is simply $\Pr[Y_i = \pm 1]$, i.e. the probability that coefficient Y_i is nonzero, or equivalently, the probability that index i is selected among the w elements of the support of Y, which happens with probability exactly w/n. Thus, $\mathbf{\Sigma}_w = \frac{w}{n}\mathbf{I}_n$ as required. The computation $\mathbf{\Sigma}_w^0$ follows similarly. □

From Lemma 1, we deduce that $\bar{\mathbf{u}} = \bar{\mathbf{y}}_1 + (\bar{\mathbf{c}}_1', \bar{\mathbf{c}}_2, \ldots, \bar{\mathbf{c}}_\ell)^\intercal$ behaves like $\mathcal{N}(0, \frac{1}{s}\mathbf{\Sigma}_{\mathbf{u}})$, where:

$$\begin{aligned}\mathbf{\Sigma}_{\mathbf{u}} &= \mathbf{\Sigma}_t \otimes \mathbf{I}_\ell + \operatorname{diag}(\mathbf{\Sigma}_{\tau-1}^0, \mathbf{\Sigma}_\tau, \ldots, \mathbf{\Sigma}_\tau) \\ &= \operatorname{diag}\Big(\frac{t}{n}, \underbrace{\frac{t}{n} + \frac{\tau-1}{n-1}, \ldots, \frac{t}{n} + \frac{\tau-1}{n-1}}_{n-1 \text{ entries}}, \underbrace{\frac{t+\tau}{n}, \ldots, \frac{t+\tau}{n}}_{n(\ell-1) \text{ entries}}\Big).\end{aligned}$$

In particular, we have:

$$\mathbf{\Sigma}_{\mathbf{u}} \prec \frac{t+\tau}{n}\mathbf{I}_{n\ell}.$$

As a result, the error term $\bar{\mathbf{e}} = \bar{\mathbf{z}} - \mathbf{G}_1 = \mathbf{G} \cdot \bar{\mathbf{u}}$ behaves like $\mathcal{N}(0, \frac{1}{s}\mathbf{\Sigma}_{\mathbf{e}})$, where $\mathbf{\Sigma}_{\mathbf{e}}$ is given by,

$$\mathbf{\Sigma}_{\mathbf{e}} = \mathbf{G}\mathbf{\Sigma}_{\mathbf{u}}\mathbf{G}^\intercal \prec \frac{t+\tau}{n}\mathbf{G}\mathbf{G}^\intercal \prec \frac{t+\tau}{n}\sigma_{\mathbf{G}}^2\mathbf{I}_{n\ell},$$

where $\sigma_{\mathbf{G}}$ denotes the largest singular value of $\mathbf{G}^\intercal$. Heuristically ignoring the inversibility condition on $\mathbf{G}$, it is sampled during key generation as a random matrix in $\mathbb{Z}^{n\ell \times n\ell}$ with i.i.d. entries having mean 0, variance $\eta_G(\eta_G + 1)/3$ and bounded fourth moment. Therefore, Bai–Yin's law [BY93, Theorem 1] shows that $\sigma_{\mathbf{G}}^2 = \big(4 + o(1)\big)n\ell \cdot \eta_G(\eta_G + 1)/3$. Hence:

$$\mathbf{\Sigma}_{\mathbf{e}} \precsim \frac{4\ell}{3}(t+\tau)\eta_G(\eta_G + 1).$$

Since, by classical concentration results,[3] a spherical Gaussian vector with covariance $\sigma^2\mathbf{I}_d$ has an infinity norm of $\sigma\sqrt{\big(2 + o(1)\big)\log d}$, we obtain:

$$\|\bar{\mathbf{e}}\|_\infty \lesssim \sqrt{\frac{8\ell}{3s}(t+\tau)\eta_G(\eta_G + 1)\log(n\ell)}.$$

Now, in order to successfully recover $\mathbf{G}_1$ by rounding $\bar{\mathbf{z}}$, it is required to have $\|\bar{\mathbf{e}}\|_\infty < 1/2$, which is satisfied when

$$s \gtrsim \frac{32\ell}{3}(t+\tau)\eta_G(\eta_G + 1)\log(n\ell).$$

[3] For example, using [AS64, 7.1.24], it is readily proven for $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I}_d)$ that the probability of having $\|\mathbf{X}\|_\infty > \sqrt{2C\log d}$ is at most $1/d^{C-1}\sqrt{\pi \log d}$ for any $C \geq 1$.

Table 3. Number of recovered coefficients of $\mathbf{G}_{1,1}$ using the basic attack of Section 3 (average over 10 repetitions of the attack).

Number of signatures S	Security level 3	Security level 5
250,000	1018.8	965.1
500,000	1023.9	1009.0
750,000	1024.0	1018.3
1,000,000	1024.0	1022.6
1,250,000	1024.0	1023.1
1,500,000	1024.0	1024.0

Furthermore, since we have restricted our attention to signatures having $\mathbf{c}_{1,0}^{(j)} = 1$, which is satisfied by a correctly generated signature with probability $\tau/2n$, obtaining those s samples satisfying the condition requires collecting a grand total of S (unconditional) signatures, where

$$S \gtrsim \frac{64n\ell}{3\tau}(t+\tau)\eta_G(\eta_G+1)\log(n\ell). \tag{6}$$

Concretely, the analysis predicts you need to see $S \approx 1.4 \cdot 10^6$ signatures to correctly recover $\mathbf{G}_1$ for the level 3 parameter set, and $S \approx 3.8 \cdot 10^6$ signatures for the level 5 parameter set.

Experimental results. This attack is implemented as code that directly calls the EagleSign-V1 signature scheme to generate a certain number of signatures, and uses them to recover the matrix $\mathbf{G}$ (or rather, the top left ring element $\mathbf{G}_{1,1}$ in the case of the level 5 parameter set which has $\ell > 1$; recovery of the other coefficients obviously proceeds in the same way). The code can be found in the file `eaglesign_ref/test/test_attack.c` within the publicly accessible repository

https://github.com/mti/attack_eaglesign.

Experimental results, collected in Table 3, show how many coefficients of the ring element $\mathbf{G}_{1,1}$, out of a total of $n = 1024$, were recovered on average over 10 runs of the attack. We can see that 500,000 to 750,000 signatures were enough to consistently recover $\mathbf{G}_{1,1}$ correctly for level 3 parameters, and 1,250,000 to 1,500,000 for level 5 parameters.

This confirms that the analysis above gives the correct order of magnitude for the number of signatures needed for recovery, but suggests that the precise estimate is somewhat pessimistic in the attack. This is most likely due to the fact that we coarsely bound $\mathbf{G}\mathbf{G}^{\mathsf{T}}$ from above by its largest singular value, while most of its other singular values are closer to the median one, which is smaller by a factor of around 4. This make our bound on $\|\bar{\mathbf{e}}\|_\infty$ not tight in most cases.

Regardless, this demonstrates that EagleSign-V1 is insecure, and can be practically broken even by a very unoptimized attack (that throws away $> 98\%$ of signature samples to begin with when filtering for the cases when $\mathbf{c}_{1,0} = 1$).

Flaw in the security proof. The fact that the scheme is vulnerable to the known message attack described above directly falsifies the claim of security against adaptively chosen message attacks in Theorem 1 of the specification document [SHDS23a]. Accordingly, the proof of that claim is flawed. This can be seen for example in part B of the proof, where the vector $\mathbf{z}$ is simulated as uniform among vectors of infinity norm $\leq \delta$ (and $\mathbf{c}$ as sampled from B_τ^ℓ independently of $\mathbf{z}$), even though it is clear, e.g., from Eq. (3) that its distribution is far from uniform.

4 Private Key Recovery With Fewer Signatures

The attack in Section 3 only takes advantage of a small fraction of the private key leakage present in signatures, while discarding over 95% of generated signatures. In this section, we show that full key recovery is in fact possible with far fewer signatures by properly taking advantage of the available leakage.

Description of the improved attack. The basic idea is as follows: by exploiting the structure of the number ring R, we can find more leakage from Eq. (2). Once we have a valid signature for some message, we obtain some $(\mathbf{z}, \mathbf{c})$ such that Eq. (2) holds for some unknown $\mathbf{y}_1 \sim \mathcal{U}(B_t^\ell)$. It is easy to see that for any $0 \leq i < 2n$ also

$$X^i \mathbf{z} = \mathbf{G} \cdot \left(X^i \mathbf{y}_1 + X^i \mathbf{c}\right),$$

holds where the unknown $X^i \mathbf{y}_1$ also follows the uniform distribution on B_t^ℓ, because multiplication by X^i is bijective on R: its inverse is $r \mapsto X^{2n-i} r$.

Since the attack from Section 3 merely requires pairs $(\mathbf{z}, \mathbf{c})$ with $\mathbf{c}_{1,0} = 1$, if we manage to find some $i \in \{0, 1, \ldots, 2n-1\}$ such that $(X^i \mathbf{c}_1)_0 = 1$ holds, we can use $(X^i \mathbf{z}, X^i \mathbf{c})$ as a sample for that attack. In fact any $\mathbf{c} \in B_\tau^\ell$ gives exactly τ many options for i such that this condition is satisfied. To see this, let us write $\mathbf{c}_1 = \sum_{j=1}^{\tau} \varepsilon_j X^{i_j}$ with $0 \leq i_1 < i_2 < \cdots < i_\tau < n$ and $\varepsilon_j \in \{-1, 1\}$ (note the signs come from $X^n = -1$). Then for all $j = 1, \ldots, \tau$, the constant coefficient of $\varepsilon_j X^{-i_j} \mathbf{c}_1$ is indeed 1. Thus, we can indeed use the attack from Section 3 to learn $\mathbf{G}_1$, by providing it the samples $(\varepsilon_j X^{-i_j} \mathbf{z}, \varepsilon_j X^{-i_j} \mathbf{c})_{1 \leq j \leq \tau}$ when seeing a signature $(\mathbf{z}, \mathbf{c})$, with $\mathbf{c}_1 = \sum_{j=1}^{\tau} \varepsilon_j X^{i_j}$ as above.

More explicitly, given a list of signatures $(\mathbf{z}^{(j)}, \mathbf{c}^{(j)})_{1 \leq j \leq S}$, the attack provides a guess for $\mathbf{G}_1$ that is given by rounding the following quantity:

$$\frac{1}{\tau S} \sum_{j=1}^{S} \sum_{k=1}^{\tau} \varepsilon_{jk} X^{-i_{jk}} \mathbf{z}^{(j)} = \frac{1}{\tau S} \sum_{j=1}^{S} \mathbf{z}^{(j)} \cdot \left(\mathbf{c}_1^{(j)}\right)^*,$$

where ε_{jk}, i_{jk} satisfy $\left(\mathbf{c}_1^{(j)}\right)^* = \sum_{k=1}^{\tau} \varepsilon_{jk} X^{i_{jk}}$. Indeed, using Eq. (2), it follows that

$$\mathbb{E}[\mathbf{z} \cdot \mathbf{c}^*] = \mathbf{G} \cdot \mathbb{E}[\mathbf{y}_1 \mathbf{c}^* + \mathbf{c}\mathbf{c}^*] = \mathbf{G} \cdot \mathbb{E}[\mathbf{c}\mathbf{c}^*],$$

holds since $\mathbb{E}[\mathbf{y}_1 \mathbf{c}^*] = \mathbb{E}[\mathbf{y}_1]\mathbb{E}[\mathbf{c}^*] = 0$.

Lemma 2. *Given a random variable $Y \sim \mathcal{U}(B_w)$, we have $\mathbb{E}[YY^*] = w$, as an element of the ring R (or equivalently $\mathbb{E}[YY^*] = (w, 0, \ldots, 0)^{\mathsf{T}}$ regarded as a vector in $\mathbb{Z}^n$).*

Moreover, for $\mathbf{c} \sim \mathcal{U}\big(B_w^\ell\big)$, we have $\mathbb{E}[\mathbf{c}\mathbf{c}^*] = w\mathbf{I}_\ell \in R^{\ell \times \ell}$.

Proof. For $Y \sim \mathcal{U}(B_w)$, we can write over the ring R:

$$\mathbb{E}[YY^*] = \sum_{i,j=0}^{n-1} \mathbb{E}[Y_i Y_j] X^{i-j} = \sum_{i,j=0}^{n-1} (\mathbf{\Sigma}_w)_{i,j} X^{i-j},$$

so it follows from Lemma 1 that $\mathbb{E}[YY^*] = \sum_{i,j=0}^{n-1} \frac{w}{n} \delta_{ij} X^{i-j} = w$, as required.

Focusing now on the matrix $\mathbb{E}[\mathbf{c}\mathbf{c}^*] \in R^{\ell \times \ell}$, its off-diagonal entries are of the form $\mathbb{E}[YZ^*]$ for independent $Y, Z \sim \mathcal{U}(B_w)$, and hence they vanish since $\mathbb{E}[Y] = 0$. The diagonal entries, on the other hand, are precisely of the form $\mathbb{E}[YY^*]$ with $Y \sim \mathcal{U}(B_w)$, which equals w by the above. □

Using Lemma 2, we get $\frac{1}{\tau}\mathbb{E}[\mathbf{z} \cdot \mathbf{c}^*] = \mathbf{G}$. Therefore, when having enough signatures $(\mathbf{z}^{(j)}, \mathbf{c}^{(j)})_{1 \le j \le S}$, we can correctly guess $\mathbf{G}$ by rounding the following coefficientwise:

$$\frac{1}{\tau}\overline{\mathbf{z} \cdot \mathbf{c}^*} = \frac{1}{\tau S} \sum_{j=1}^{S} \mathbf{z}^{(j)} \cdot \left(\mathbf{c}^{(j)}\right)^*.$$

Analysis of the improved attack. Intuitively, compared to Eq. (6) one could expect the improved attack requires a factor $2n$ less signatures to work, as each signature now provides τ samples of the form Eq. (5) compared to the $\tau/2n$ (on average) from Section 3. Namely, a naive estimate is that the improved attack succesfully recovers $\mathbf{G}$ after seeing

$$S \gtrsim \frac{32\ell}{3\tau}(t + \tau)\eta_G(\eta_G + 1)\log(n\ell),$$

signatures, which translates to $S \approx 700$ and $S \approx 1900$ signatures for the level 3 and level 5 parameter sets respectively.

This is not very rigorous, however, because the τ samples coming from a single signature are not in fact independent, which violates the independence hypothesis of standard forms of the central limit theorem. One can obtain a more rigorous analysis by either relying on variants of the CLT that hold under weak dependence (e.g., CLT variants with α-mixing, with hypotheses that easily hold in our case), or simply by considering each signature as a sample and analyzing its behavior under standard CLT. This can be done for example based on the following somewhat tedious result.

Lemma 3. *(a) The covariance matrix $\widehat{\mathbf{\Sigma}}_{w,w'}$ of uv^* for independent $u \sim \mathcal{U}(B_w)$ and $v \sim \mathcal{U}(B_{w'})$ (with uv^* seen as vector in $\mathbb{Z}^n$) is given by:*

$$\widehat{\mathbf{\Sigma}}_{w,w'} = \frac{ww'}{n}\mathbf{I}_n.$$

(b) The covariance matrix $\widetilde{\mathbf{\Sigma}}_w$ of vv^* for $v \sim \mathcal{U}(B_w)$ (with vv^* seen as a vector in $\mathbb{Z}^n$) is given by, for all $0 \leq i, j \leq n-1$:

$$(\widetilde{\mathbf{\Sigma}}_w)_{ij} = \begin{cases} \frac{w(w-1)}{n-1} & \text{if } i = j \text{ and } i \neq 0, \frac{n}{2}; \\ -\frac{w(w-1)}{n-1} & \text{if } i + j = n \text{ and } i \neq 0, \frac{n}{2}; \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, the eigenvalues of the symmetric matrix $\widetilde{\mathbf{\Sigma}}$ are 0 with multiplicity $\frac{n}{2}+1$, and $2w(w-1)/(n-1)$ with multiplicity $\frac{n}{2}-1$. In particular, we have

$$\widetilde{\mathbf{\Sigma}}_w \prec \frac{2w(w-1)}{n-1}\mathbf{I}_n.$$

Proof. See Appendix A. □

Define for each signature the matrix $\mathbf{U}^{(j)} = \left(\mathbf{y}_1^{(j)} + \mathbf{c}^{(j)}\right) \cdot \left(\mathbf{c}^{(j)}\right)^* - \tau\mathbf{I}_\ell$, and let $\overline{\mathbf{U}} = \frac{1}{S}\sum_j \mathbf{U}^{(j)}$. The error term is given by

$$\frac{1}{\tau}\overline{\mathbf{z}\mathbf{c}^*} - \mathbf{G} = \frac{1}{\tau}\mathbf{G}\overline{\mathbf{U}}. \tag{7}$$

Now, each $\mathbf{U}^{(j)}$ has zero expected value, and in its expression, the diagonal entries of $\mathbf{c}^{(j)}\left(\mathbf{c}^{(j)}\right)^* - \tau\mathbf{I}_\ell$ have covariance $\widetilde{\mathbf{\Sigma}}_\tau$ by Lemma 3 above. The off-diagonal entries, on the other hand, have the form uv^* for independent $u, v \sim \mathcal{U}(B_\tau)$, so their covariance is $\widehat{\mathbf{\Sigma}}_{\tau,\tau} = \frac{\tau^2}{n}\mathbf{I}_n$. Similarly, all entries of $\mathbf{y}_1^{(j)} \cdot \left(\mathbf{c}^{(j)}\right)^*$ have covariance $\widehat{\mathbf{\Sigma}}_{t,\tau} = \frac{t\tau}{n}\mathbf{I}_n$.

It follows that the diagonal (resp., off-diagonal) entries of $\overline{\mathbf{U}}$ behave like Gaussian vectors $\sim \mathcal{N}(0, \frac{1}{S}\mathbf{\Sigma}_{\text{diag}})$ (resp., $\mathcal{N}(0, \frac{1}{S}\mathbf{\Sigma}_{\text{off-diag}})$) with:

$$\mathbf{\Sigma}_{\text{diag}} = \widehat{\mathbf{\Sigma}}_{t,\tau} + \widetilde{\mathbf{\Sigma}}_\tau \prec \left(\frac{t\tau}{n} + \frac{2\tau(\tau-1)}{n-1}\right)\mathbf{I}_n \prec \frac{\tau(t+2\tau)}{n}\mathbf{I}_n.$$

$$\mathbf{\Sigma}_{\text{off-diag}} = \widehat{\mathbf{\Sigma}}_{t,\tau} + \widehat{\mathbf{\Sigma}}_{\tau,\tau} = \left(\frac{t\tau}{n} + \frac{\tau^2}{n}\right)\mathbf{I}_n \prec \frac{\tau(t+2\tau)}{n}\mathbf{I}_n.$$

Therefore, the error term in Eq. (7) has entries behaving like Gaussian vectors $\mathcal{N}(0, \frac{1}{\tau^2 S}\mathbf{G}\mathbf{\Sigma}\mathbf{G}^\intercal)$ with $\mathbf{\Sigma} = \mathbf{\Sigma}_{\text{diag}}$ on the diagonal of $\bar{\mathbf{U}}$ and $\mathbf{\Sigma} = \mathbf{\Sigma}_{\text{off-diag}}$ off it. In all cases, we get a bound $\prec \frac{t+2\tau}{n\tau S}\sigma_{\mathbf{G}}^2\mathbf{I}_n$ on the covariance, which allows to conclude similarly as before. The condition on S for full recovery (taking into account that $\mathbf{G}$ has $n\ell^2$ coefficients) now becomes:

$$S \gtrsim \frac{32\ell}{3\tau}(t+2\tau)\eta_G(\eta_G+1)\log(n\ell^2),$$

which is almost the same as the naive estimate, fortunately.

Table 4. Probability that the attack of Section 4 recovers **G** and **D** successfully using the leakage in respectively Eq. (2) and Eq. (4), based on 1000 independent repetitions. Note: if **G** is successfully recovered, one can also successfully recover **D** by using Eq. (1) instead of the leakage in Eq. (4).

Number of signatures S	Security level 3		Security level 5	
	G	**D**	**G**	**D**
300	100.0%	0.0%	0.2%	0.0%
500	100.0%	0.0%	81.6%	0.0%
2000	100.0%	69.3%	100.0%	97.7%

Experimental results. Table 4 contains the number of signatures that was required to perform a complete private key recovery attack. The key recovery attack can be found in the file `code/key_recovery.c` within the publicly accessible repository

https://github.com/ludopulles/EagleHasFlown.

One private key recovery attack with 2000 signatures runs in 2 seconds on a single core of an Intel i5-4590 CPU.

Note that Table 4 shows that it is more efficient to recover **D** using Eq. (1) with the recovered **G**, rather than performing a similar attack on the leakage of **D** in Eq. (4) directly.

Although Section 3 already shows that EagleSign-V1 is not secure, we have shown that there is a very practical and efficient key recovery attack, requiring only 500 messages and signatures, from which we can conclude that the scheme needs to be altered significantly to have hope of security.

5 A Known-Message Attack on EagleSign-V2

We now show how the same attack strategy from Section 4 basically extends to EagleSign-V2 [SHDS23b], a scheme introduced as a "fixed" variant of EagleSign-V1 in response to the previous attack. EagleSign-V2 is claimed to satisfy the zero-knowledge property that signatures are independent of the secret key, but actually does not.

Attack strategy. A first observation is that, like in EagleSign-V1, all the ring elements in the expression for $\mathbf{z} = g \cdot (\mathbf{y} + \mathbf{Fc})$ are small, and therefore the equality actually holds over the ring (there is no modular reduction).

The rejection sampling step (Line 6 of Algorithm 2) introduced by the submitters to make their scheme "zero-knowledge" rejects vectors $\mathbf{z}$ such that $\|\mathbf{z}\|_\infty > t_g(\gamma_1 - \ell\tau)$, where t_g is the ℓ_1-norm of ternary ring element g, γ_1 bounds $\|\mathbf{y}\|_\infty$ and τ is the ℓ_1-norm of ternary element $\mathbf{c}$. But it is easy to see that this rejection condition is essentially never triggered for the proposed parameters!

This is because the coefficients of $g\mathbf{y}$ (which are much larger than those of $g\mathbf{Fc}$, and hence dominate in $\mathbf{z}$) are each distributed as a sum of t_g independent integers sampled from $\mathcal{U}([-\gamma_1, \gamma_1])$, so they are sub-Gaussian with standard deviation roughly $\gamma_1\sqrt{t_g/3}$. To trigger the bound, they need to reach > 4.5 standard deviations for level 2 parameters and > 6.4 standard deviations for level 5 parameters. These events are too rare to measurably affect the distribution of elements $\mathbf{z}$, and therefore, from a statistical standpoint, everything happens as if the rejection sampling step was removed entirely.

There is another restart condition in the signing generation Algorithm 2: namely Line 9, which is meant for correctness rather than security. This step *can* be triggered with noticeable probability, but again should not affect the distribution of $\mathbf{z}$. This is due to the fact that $\mathbf{v} = \mathbf{Ez} - \mathbf{Ac} \bmod qR^k$ behaves essentially as the image of $(\mathbf{z}, \mathbf{c})$ under a hash function, and hence does not meaningfully depend on $\mathbf{z}$.

Thus, for statistical purposes, EagleSign-V2 signature elements $\mathbf{z}$ are distributed as samples of the form $g \cdot (\mathbf{y} + \mathbf{Fc})$ for independent $\mathbf{y} \sim \mathcal{U}\big(S_{\gamma_1}^{\ell}\big)$ and $\mathbf{c} \sim \mathcal{U}\big(B_{\tau}^{\ell}\big)$.

Universal forgery attack. Therefore, we can apply the same strategy as in Section 4. We collect many signatures, and recover by averaging the expectation of $\mathbf{z} \cdot \mathbf{c}^*$, which in this case is given by:

$$\mathbb{E}[\mathbf{zc}^*] = g \cdot \mathbb{E}[\mathbf{yc}^* + \mathbf{Fcc}^*] = 0 + g\mathbf{F} \cdot \mathbb{E}[\mathbf{cc}^*] = \tau \cdot g\mathbf{F},$$

by Lemma 2.

Now, this recovered matrix $g\mathbf{F}$ is already sufficient to forge signatures on arbitrary messages. Indeed, one can for example sign with randomness $\mathbf{y} = 0$, which amounts to forge the signature $(r, \mathbf{z}) = \big(\mathcal{G}(0)\,, g\mathbf{F} \cdot \mathcal{H}(m, r)\,\big)$ for message m. Because

$$\mathbf{Ez} - \mathbf{A} \cdot \mathcal{H}(m, r) = \mathbf{DF} \cdot \mathcal{H}(m, r)$$

is short, its high bits are zero, which makes this signature valid.

More generally, suppose we are given a signature $(r_0, \mathbf{z}_0)$ on a message m_0, so that $\mathbf{z}_0 = g \cdot (\mathbf{y}_0 + \mathbf{Fc}_0)$ for $\mathbf{c}_0 = \mathcal{H}(m_0, r_0)$ and some unknown randomness vector $\mathbf{y}_0$. Now with good probability, we can use the knowledge of $g\mathbf{F}$ to forge a valid signature for any arbitrary message $m \neq m_0$ with the same implied randomness $\mathbf{y}_0$ by setting:

$$\begin{aligned} &r = r_0, \\ &\mathbf{c} = \mathcal{H}(m, r) = \mathcal{H}(m, r_0)\,, \\ &\mathbf{z} = \mathbf{z}_0 + g\mathbf{F} \cdot (\mathbf{c} - \mathbf{c}_0) = g \cdot (\mathbf{y}_0 + \mathbf{Fc}). \end{aligned}$$

Then $(r, \mathbf{z})$ will be a valid signature for m if it implicitly passes the correctness check of Line 9 of Algorithm 2, which happens with good probability. Thus, we already get a universal forgery attack.

Full key recovery. It is possible to turn the previous attack into a full (equivalent) key recovery attack by recovering the ring element g using algebraic techniques. Clearly, once g is recovered, one also obtains $\mathbf{F} = g^{-1} \cdot (g\mathbf{F})$ and $\mathbf{D} = g\mathbf{E} - \mathbf{A}\mathbf{F}^{-1}$.

We propose to recover g (up to some power of X) as follows. This approach, while not asymptotically efficient, is practical at least for level 2 parameters, and easily invalidates the security claims of all parameter sets.

One can first observe that the ideal $gR \subset R$ is easy to recover from a few signature elements $\mathbf{z}^{(j)} = g \cdot (\mathbf{y}^{(j)} + \mathbf{F}\mathbf{c}^{(j)}) = g \cdot \mathbf{u}^{(j)}$. Indeed, all the entries of each $\mathbf{z}^{(j)}$ are in the ideal, and they generate it as soon as the entries of all the $\mathbf{u}^{(j)}$ are setwise coprime in R. Let's say we have s signatures, and hence ℓs entries in total. Since these elements are sufficiently random (in the sense that they are not expected to satisfy any congruence condition with unusually high or unusually low probability), the setwise coprimality condition heuristically happens with probability $1/\zeta_K(\ell s)$ where ζ_K is the Dedekind zeta function of the cyclotomic field K which is the fraction field of R.[4] Note that $1/\zeta_K(2) \approx 3/4$ and $\zeta_K(t)$ very quickly converges to 1 as t increases, so just a handful of signatures are sufficient to exactly recover gR with high probability.

It is a classical fact about circular rotations of a sparse vector, that there exists a root of unity $\omega = \pm X^r \in R$ such that ωg has constant coefficient 1, and such that among the t_g nonzero coefficients of ωg, exactly $\lceil t_g/2 \rceil$ occur in the first $n/2$ coefficients. Thus, by replacing g by ωg we can assume, without loss of generality, that g is of the form $1+g_1+g_2$ where $g_1, g_2 \in R$ are ternary, of ℓ_1-norm $t_1 = \lceil t_g/2 \rceil - 1$ and $t_2 = \lfloor t_g/2 \rfloor$ respectively, and with support $\{1, \ldots, n/2-1\}$ and $\{n/2, \ldots, n-1\}$ respectively.

We can thus recover g by a meet-in-the-middle attack: form a sorted list of the residues[5] mod gR of the $2^{t_1}\binom{n/2-1}{t_1}$ candidates for g_1, and then check by binary search for each of the $2^{t_2}\binom{n/2}{t_2}$ candidates for g_2 if $-1 - g_2 \bmod gR$ matches on the candidates for g_1. For matching pairs (g_1, g_2), we can easily check by a norm computation if $1 + g_1 + g_2$ generates the ideal gR. The overall algorithm runs in time and space complexity $\widetilde{O}\left(2^{t_2}\binom{n/2}{t_2}\right)$, and by construction outputs a generator g' of gR, which is thus necessarily of the form $g' = \nu g$ for some unit ν of R, and is also in B_{t_g}. Since νg for a unit which is *not* a root of unity is overwhelmingly likely to not be ternary or have Hamming weight t_g, the output should be g up to multiplication by a root of unity. This is enough to obtain an equivalent key to the actual signing key.

The complexity is around $2^{27.4}$ for the level 2 parameter set and $2^{64.7}$ for the level 5 parameter set. The former is practical, and while the latter is probably difficult to mount in practice, it is still far below the 256-bit security level claimed by the authors.

[4] This follows easily from the Euler product expansion of ζ_K.

[5] Since we represent the ideal gR by its Hermite normal form, R/gR is simply described as a product $\mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_r\mathbb{Z}$ where $\prod d_i$ is the algebraic norm of g. Thus, the reduction mod gR of an element of R is concretely represented by the list of residues modulo each of the d_i's in that decomposition.

Analysis of the forgery attack. The analysis of the statistical attack is essentially the same as in Section 4. Given S signature samples $(\mathbf{z}^{(j)}, \mathbf{c}^{(j)})$, we define $\mathbf{U}^{(j)} = \mathbf{y}_1^{(j)} \cdot \left(\mathbf{c}^{(j)}\right)^* + \mathbf{F}\left(\mathbf{c}^{(j)}\left(\mathbf{c}^{(j)}\right)^* - \tau \mathbf{I}_\ell\right)$, and let $\overline{\mathbf{U}} = \frac{1}{S}\sum_j \mathbf{U}^{(j)}$. Similar to Eq. (7), we have:

$$\frac{1}{\tau}\overline{\mathbf{z}\mathbf{c}^*} - g\mathbf{F} = \frac{1}{\tau} g\overline{\mathbf{U}}. \tag{8}$$

Now, each $\mathbf{U}^{(j)}$ has zero expected value, and the same reasoning as before shows that its diagonal (resp., off-diagonal) entries of $\overline{\mathbf{U}}$ behave like Gaussian vectors $\sim \mathcal{N}(0, \frac{1}{S}\boldsymbol{\Sigma}_{\text{diag}})$ (resp., $\mathcal{N}(0, \frac{1}{S}\boldsymbol{\Sigma}_{\text{off-diag}})$) with:

$$\begin{aligned}
\boldsymbol{\Sigma}_{\text{diag}} &= \widehat{\boldsymbol{\Sigma}}'_{\gamma_1,\tau} + \mathbf{F}\widetilde{\boldsymbol{\Sigma}}_\tau \mathbf{F}^\mathsf{T} \prec \widehat{\boldsymbol{\Sigma}}'_{\gamma_1,\tau} + \frac{2\tau^2}{n}\sigma_{\mathbf{F}}^2 \mathbf{I}_n \\
\boldsymbol{\Sigma}_{\text{off-diag}} &= \widehat{\boldsymbol{\Sigma}}'_{\gamma_1,\tau} + \mathbf{F}\widehat{\boldsymbol{\Sigma}}_{\tau,\tau} \mathbf{F}^\mathsf{T} \prec \widehat{\boldsymbol{\Sigma}}'_{\gamma_1,\tau} + \frac{\tau^2}{n}\sigma_{\mathbf{F}}^2 \mathbf{I}_n,
\end{aligned}$$

where $\widetilde{\boldsymbol{\Sigma}}$ and $\widehat{\boldsymbol{\Sigma}}$ are defined as before, $\sigma_{\mathbf{F}}^2$ is the top singular value of $\mathbf{F}$, and $\widehat{\boldsymbol{\Sigma}}'_{\beta,w}$ is the covariance matrix of a product uv^* with independent $u \sim \mathcal{U}(S_\beta)$ and $v \sim \mathcal{U}(B_w)$. The proof of Lemma 3(a) is readily adapted to show that:

$$\widehat{\boldsymbol{\Sigma}}'_{\beta,w} = \frac{w\beta(\beta+1)}{3}\mathbf{I}_n.$$

and Bai–Yin's law shows again that $\sigma_{\mathbf{F}}^2 \lesssim 4n\ell \cdot \eta_f(\eta_f+1)/3$. As before, the error term in Eq. (8) has entries behaving like Gaussian vectors $\mathcal{N}(0, \frac{1}{\tau^2 S}\mathbf{M}_g \boldsymbol{\Sigma}\mathbf{M}_g^\mathsf{T})$ with $\mathbf{M}_g$ the matrix corresponding to multiplication by g on vectors of R^ℓ; and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{diag}}$ on the diagonal of $\bar{\mathbf{U}}$, resp. $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\text{off-diag}}$ off it. In all cases, we get a bound $\prec \frac{B}{\tau S}\sigma_{\mathbf{M}_g}^2 \mathbf{I}_n$ on the covariance, where $\sigma_{\mathbf{M}_g}^2$ is the top singular value of $\mathbf{M}_g$ and:

$$B = \frac{\gamma_1(\gamma_1+1)}{3} + \frac{2\tau}{n}\sigma_{\mathbf{F}}^2 \lesssim \frac{\gamma_1(\gamma_1+1) + 8\tau\ell\cdot\eta_f(\eta_f+1)}{3} \approx \frac{\gamma_1^2}{3}.$$

On the other hand, $\mathbf{M}_g = \operatorname{diag}(g, \dots, g)$ when regarded as a matrix in $R^{\ell\times\ell}$, so its top singular value is the top singular value of multiplication by g, seen as an operator on $\mathbb{Z}^n$, which is the maximum of the squared absolute values of its Fourier coefficients. Since g is ternary of Hamming weight t_g, this is bounded by t_g^2. Hence the bound on the covariance of the coefficients in Eq. (8) becomes $B \cdot t_g^2/(\tau S)$. Therefore, we expect to fully recover $g\mathbf{F}$ with S signatures when:

$$S \gtrsim \frac{4t_g^2\gamma_1^2}{3\tau}\log(n\ell^2).$$

This suggests that $S \gtrsim 8 \cdot 10^9$ (around 2^{32}) signatures are sufficient to fully recover $g\mathbf{F}$ for the level 2 parameters, and $S \gtrsim 2.5 \cdot 10^{11}$ (around 2^{38}) for the level 5 parameters. However, this is an even more pessimistic analysis than the one of Section 3, since most singular values of $\mathbf{M}_g$ are closer to t_g than t_g^2 (so we may be overestimating S by a factor close to t_g instead of ≈ 4 previously).

Improvement if the algebraic attack is carried out first. In case we manage to recover the secret key element g first using the algebraic techniques of the previous paragraph "Full key recovery", we can recover $\mathbf{F}$ using the same approach with substantially fewer signatures. Indeed, we can then take $1/\tau$ times the mean of the samples $g^{-1} \cdot \mathbf{z}^{(j)}\left(\mathbf{c}^{(j)}\right)^*$, and the error term becomes:

$$\frac{1}{\tau}\overline{g^{-1} \cdot \mathbf{zc}^*} - \mathbf{F} = \frac{1}{\tau}\overline{\mathbf{U}}.$$

which is bounded in the same way as before, except that there is no multiplication by $\mathbf{M}_g$. Therefore, we expect to fully recover $\mathbf{F}$ with S signatures using this approach when:

$$S \gtrsim \frac{4\gamma_1^2}{3\tau}\log(n\ell^2).$$

This suggests that $S \gtrsim 1.6 \cdot 10^8$ (around 2^{27}) signatures are sufficient to fully recover $\mathbf{F}$ for the level 2 parameters, and $S \gtrsim 1.5 \cdot 10^9$ (around 2^{30}) for the level 5 parameters. This analysis does not involve coarse overestimates of singular values, so it should be fairly tight.

Experimental results. We have implemented the attack, which can be found in the file `code/key_recovery.c` within the publicly accessible repository

https://github.com/mti/Eagle2HasFlown.

Since the signature scheme is relatively slow, the attack is parallelized using OpenMP and ran on a relatively large workstation with a 48-core/96-thread Xeon Platinum CPU.

For the level 2 parameters, we ran the attack on 10 sets of $S = 3 \cdot 10^9$ signatures, and achieved full recovery of $g\mathbf{F}$ in all cases. On our workstation, each set takes just above 18 hours on the wall clock, dominated by the cost of generating the required signatures. The behavior of the key recovery is displayed on the

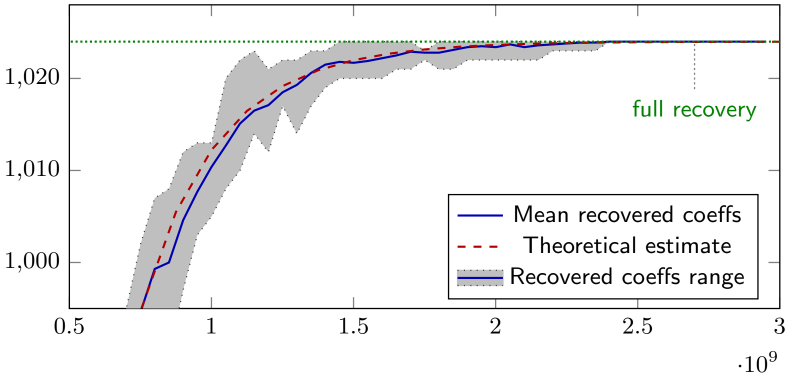

Fig. 1. Attack on 10 instances of EagleSign-V2 level 2.

graph of Fig. 1: the top and bottom boundaries of the gray region represent the minimum and maximum number of coefficients recovered among the 10 experiments for the number of signatures given on the x-axis, with the mean shown in the thick continuous line in the middle. The dashed line represents the expected number of recovered coefficients for the given number of signatures, assuming that the error term is Gaussian with variance $\frac{Bt_g}{\tau S}$ (as we expect heuristically), instead of the more pessimistic $\frac{Bt_g^2}{\tau S}$. As can be seen on the graph, the heuristic estimate models the actual behavior of the attack very closely.

Moreover, in the file `code/key_recoveryF.c` within the same repository, we have implemented a variant of the attack which assumes that g itself has already been recovered using the algebraic techniques from the paragraph "Full key recovery", which in this case would be less expensive than simply generating the signature batch for the statistical attack. It then applies the statistical attack on the values

$$g^{-1} \cdot \mathbf{z}\mathbf{c}^* = (\mathbf{y} + \mathbf{F}\mathbf{c})\,\mathbf{c}^*,$$

in order to recover $\mathbf{F}$. In the analysis, this improves the attack by a factor t_g.

For the level 2 parameters, we ran the attack on 10 sets of $S = 4 \cdot 10^8$ signatures, and achieved full recovery of $\mathbf{F}$ in all cases. On our workstation, each set takes about 160 minutes on the wall clock, again dominated by the cost of generating the required signatures. The behavior of the key recovery is displayed on the graph of Fig. 2 in the same format as above. Again, our estimate models the actual behavior of the attack very closely.

Flaw in the security proof. One flaw in the security proof of EagleSign-V2 occurs in [SHDS23b, §3.2], which incorrectly claims that for all choices of $(\mathbf{z}, \mathbf{c})$ satisfying the rejection condition, the conditional probability $\Pr\left[g\mathbf{y} = \mathbf{z} - g\mathbf{F}\mathbf{c} \,\middle|\, \mathbf{c}\right]$ is the same. However, this is clearly false, because the distribution of $g\mathbf{y}$ over the interval $[-t_g\gamma_1, t_g\gamma_1]$ is far from uniform. Namely, its coefficients

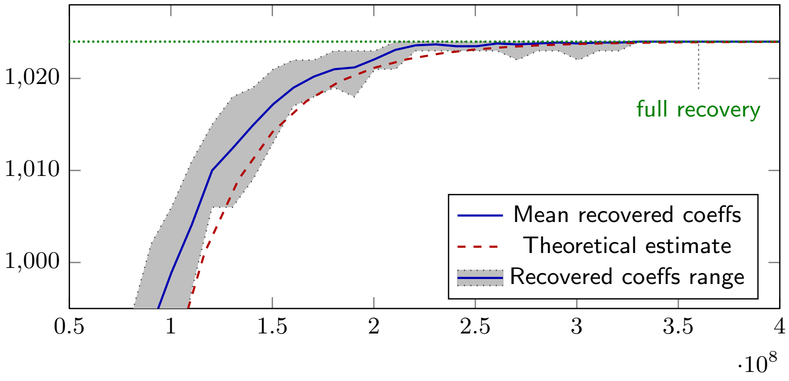

Fig. 2. Attack on 10 instances of EagleSign-V2 level 2, after prior recovery of g.

concentrate around 0, so that, e.g., vectors $\mathbf{z}$ with coefficients close to the bound occur with much lower probability than smaller ones.

6 Conclusion

In Section 3 and Section 4 we have demonstrated that the scheme EagleSign-V1 suffers from key leakage, by practically providing an attack that performs a full key recovery attack, requiring just a few hundred signatures. We also identified a flaw in the provided security proof.

Moreover, the subsequent EagleSign-V2 proposal, aimed at fixing these issues, fails to do so: indeed, we have shown in Section 5 that a largely similar vulnerability persists in the new scheme. Although exploiting it in practice requires several orders of magnitude more signatures that in EagleSign-V1, the scheme can be broken with far fewer than the 2^{64} signatures required by NIST [NIST23, 4.B.2]. We again also show that the security proof is incorrect.

These findings underscore the need for sound security analysis and reliance on serious frameworks, such as Lyubashevsky's Fiat–Shamir with aborts.

References

AS64. M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. U.S. Government Printing Office, 1964. https://archive.org/details/AandS-mono600.

BBE+19. G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, M. Rossi, and M. Tibouchi. GALACTICS: Gaussian sampling for lattice-based constant- time implementation of cryptographic signatures, revisited. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 2147–2164. ACM Press, Nov. 2019. .

BY93. Z. D. Bai and Y. Q. Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. *The Annals of Probability*, 21(3):1275–1294, 1993. .

CPQC. Chinese post-quantum cryptography competition. Technical report, Chinese Association for Cryptologic Research, 2020. available at https://www.cacrnet.org.cn/site/content/854.html.

dEK+23. R. del Pino, T. Espitau, S. Katsumata, M. Maller, F. Mouhartem, T. Prest, M. Rossi, and M. Saarinen. Raccoon. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

dPRS23. R. del Pino, T. Prest, M. Rossi, and M.-J. O. Saarinen. High-order masking of lattice signatures in quasilinear time. In *2023 IEEE Symposium on Security and Privacy*, pages 1168–1185. IEEE Computer Society Press, May 2023. .

EFGT17. T. Espitau, P.-A. Fouque, B. Gérard, and M. Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1857–1874. ACM Press, Oct. / Nov. 2017. .

HBD+22. A. Hülsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, J.-P. Aumasson, B. Westerbaan, and W. Beullens. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

Hou23. A. C. Hounkpev. Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign-V2, A New improvement of EagleSign with Zero Knowledge property, 2023. available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/Nqb51xHtzGE/m/awgsea9yAQAJ.

KPQC. Korean quantum resistant cryptography national contest. Technical report, Quantum Resistant Cryptography Research Center, 2021. available at https://kpqc.or.kr/contents/03_exhibit/board.html?board_id=board_competition&mode=view&no=6&cate=.

LDK+22. V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, and S. Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

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

NIST17. Post-quantum cryptography standardization. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

NIST23. Standardization of additional digital signature schemes. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/standardization.

PFH$^+$22. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

Pul23. L. Pulles. Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign, 2023. available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/zas5PLiBe6A/m/p_pevgyFBQAJ.

SHDS23a. D. Sow, A. C. Hounkpevi, S. Djimnaibeye, and M. Seck. EagleSign. Technical report, National Institute of Standards and Technology, 2023. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures.

SHDS23b. D. Sow, A. C. Hounkpevi, S. Djimnaibeye, and M. Seck. EagleSign-V2 : A secure instantiation of EagleSign which is an ElGamal pq-signature over lattices, 2023. available at https://github.com/eaglesignteam/eaglesign_v2/blob/master/Supporting_Documentation/EagleSignV2-NIST-10-December-2023.pdf.

Tib23. M. Tibouchi. Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign, 2023. available at https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/zas5PLiBe6A/m/A2KSHtqUAgAJ.

Appendix

A Proof of Lemma 3

Part (a). Since uv^* has zero expected value, the covariance matrix $\widehat{\mathbf{\Sigma}}_{w,w'}$ has simply the value $\mathbb{E}[(uv^*)_i(uv^*)_j]$ at entry (i,j). Now with the convention that $u_{k+n} \coloneqq -u_k$ for $0 \leq k < n$, one can show $(uv^*)_i = \sum_{k=0}^{n} u_{k+i}v_k$. Now by independence of u and v, we have:

$$\mathbb{E}\Big[(uv^*)_i(uv^*)_j\Big] = \sum_{k,\ell=0}^{n-1} \mathbb{E}[u_{k+i}u_{\ell+j}]\cdot\mathbb{E}[v_k v_\ell] = \sum_{k,\ell=0}^{n-1} \frac{w}{n}\delta_{k+i,\ell+j}\cdot\frac{w'}{n}\delta_{k\ell} = \frac{ww'}{n^2}\cdot n\delta_{ij},$$

which proves part (a).

Part (b). By Lemma 2, we have $\mathbb{E}[vv^*] = (w,0,\ldots,0)^\mathsf{T}$ as an element of R. By writing $\mathbf{M} = \mathbb{E}[(vv^*)(vv^*)^\mathsf{T}]$, we have

$$\widetilde{\mathbf{\Sigma}}_w = \mathbb{E}[(vv^*)(vv^*)^\mathsf{T}] - \mathbb{E}[vv^*]\mathbb{E}[vv^*]^\mathsf{T} = \mathbf{M} - \operatorname{diag}(w^2,0,\ldots,0),$$

so it suffices to compute $\mathbf{M}$.

First, let us compute $\mathbf{M}_{0j}$. For any $v \in B_w$, we have $(vv^*)_0 = \|v\|_2^2 = w$ so $\mathbf{M}_{0j} = w \cdot \mathbb{E}[(vv^*)_j] = w^2\delta_{0,j}$ by Lemma 2.

Note that for any $v \in B_w$, we have $vv^* = (vv^*)^*$ so $(vv^*)_{n/2} = 0$ and $(vv^*)_i = -(vv^*)_{n-i}$ holds for $1 \leq i < n$, which implies $\mathbf{M}_{i,n/2} = 0$ and $\mathbf{M}_{n-i,j} = -\mathbf{M}_{i,j}$ for $1 \leq i < n$. Because $\mathbf{M}$ is a symmetric matrix, to compute $\widetilde{\mathbf{\Sigma}}_w$ it suffices to compute

$$\mathbf{M}_{ij} = \mathbb{E}[(vv^*)_i(vv^*)_j] = \sum_{k,\ell=0}^{n-1} \mathbb{E}[v_k v_{k+i} v_\ell v_{\ell+j}],$$

for all $0 \leq i \leq j < n/2$.

For the other $\mathbf{M}_{ij}$ with $1 \leq i \leq j < n/2$, note when one of $\{k, k+i, \ell, \ell+j\}$ is different from the other three (modulo n), then $\mathbb{E}[v_k v_{k+i} v_\ell v_{\ell+j}] = 0$. For example, if $k \notin \{k+i, \ell, \ell+j\}$, then $\Pr[v_k v_{k+i} v_\ell v_{\ell+j} = 1] = \Pr[v_k v_{k+i} v_\ell v_{\ell+j} = -1]$ by the independence of the choices of the signs, similar to the proof of Lemma 1.

Now let us suppose $\mathbb{E}[v_k v_{k+i} v_\ell v_{\ell+j}] \neq 0$ for some $1 \leq i \leq j < n/2$ and $0 \leq k, \ell < n$. This necessarily means that either $k = \ell$ or $k \equiv \ell+j \pmod{n}$ holds. In this second case $k \equiv \ell + j$, then we also need $k + i \in \{k, \ell, \ell+j\} = \{k, k-j\}$. $k + i = k$ cannot hold since $i \neq 0$, and $k + i \equiv k - j \pmod{n}$ cannot hold because $0 < i + j < n$. Therefore, we necessarily have $k = \ell$. Additionally $k + i \in \{k, \ell, \ell+j\} = \{k, k+j\}$ is required, which must mean $k + i = k + j$, or equivalently $i = j$. Hence, $\mathbf{M}_{ij}$ vanishes for all $1 \leq i < j < n/2$, while $\mathbf{M}_{ii} = \sum_k \mathbb{E}[v_{k+i}^2 v_k^2]$ for $1 \leq i < n/2$.

Observe that $\mathbb{E}[v_k^2 v_{k+i}^2]$ equals the probability that both v_k and v_{k+i} are nonzero, which happens if and only if $\{k, k+i\}$ is a subset of the support of v. This happens with probability exactly:

$$\mathbb{E}[v_k^2 v_{k+i}^2] = \frac{\binom{n-2}{w-2}}{\binom{n}{w}} = \frac{(n-2)!w!(n-w)!}{(w-2)!(n-w)!n!} = \frac{w(w-1)}{n(n-1)}.$$

By summing over k, we obtain $\mathbf{M}_{ii} = w(w-1)/(n-1)$ for $1 \leq i < n/2$.

This all gives the required form for $\widetilde{\boldsymbol{\Sigma}}_w$, which can be written as $\widetilde{\boldsymbol{\Sigma}}_w = \frac{w(w-1)}{n-1} \cdot \mathbf{N}$, where

$$\mathbf{N} = \begin{pmatrix} 0 & 0 & & & \cdots & & & 0 \\ 0 & 1 & & & \cdots & & & -1 \\ & & \ddots & & & & \iddots & \\ & & & 1 & 0 & -1 & & \\ \vdots & & & 0 & 0 & 0 & & \\ & & & -1 & 0 & 1 & & \\ 0 & & \iddots & & & & \ddots & \\ 0 & -1 & & & \cdots & & & 1 \end{pmatrix}.$$

To conclude, we simply need to compute the eigenvalues of $\widetilde{\boldsymbol{\Sigma}}$, which amounts to finding the eigenvalues of $\mathbf{N}$. Remark now that $\mathbf{N}$ is isometrically conjugate (equal up to a permutation of the rows and columns) to $\mathrm{diag}(\mathbf{A}, \mathbf{B}, \ldots, \mathbf{B})$, where

$$\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

The eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ are $\{0, 0\}$ and $\{0, 2\}$ respectively with multiplicities, which completes the proof.