Research Article

Mehdi Tibouchi* and Alexandre Wallet

One Bit is All It Takes: A Devastating Timing Attack on BLISS's Non-Constant Time Sign Flips

https://doi.org/10.1515/jmc-2020-0079
Received Jun 05, 2019; accepted Jul 01, 2019

Abstract: As one of the most efficient lattice-based signature schemes, and one of the only ones to have seen deployment beyond an academic setting (e.g., as part of the VPN software suite strongSwan), BLISS has attracted a significant amount of attention in terms of its implementation security, and side-channel vulnerabilities of several parts of its signing algorithm have been identified in previous works. In this paper, we present an even simpler timing attack against it. The bimodal Gaussian distribution that BLISS is named after is achieved using a random sign flip during signature generation, and neither the original implementation of BLISS nor strongSwan ensure that this sign flip is carried out in constant time. It is therefore possible to recover the corresponding sign through side-channel leakage (using, e.g., cache attacks or branch tracing). We show that obtaining this single bit of leakage (for a moderate number of signatures) is in fact sufficient for a full key recovery attack. The recovery is carried out using a maximum likelihood estimation on the space of parameters, which can be seen as a statistical manifold. The analysis of the attack thus reduces to the computation of the Fisher information metric.

Keywords: Lattice-Based Cryptography, Cryptanalysis, BLISS, Side-Channel Analysis, Maximum Likelihood Estimation, Parametric Inference, Information Geometry

2010 Mathematics Subject Classification: 94A60, 62F30, 60D05

1 Introduction

While lattice-based cryptography is a mature research area from a theoretical standpoint, the study of implementation issues related to lattice-based schemes is more recent. In particular, *implementation attacks* against such schemes and their countermeasures have only emerged as a subject of active research in the past few years, motivated in large part by the push towards postquantum cryptography standardization and deployment epitomized by the NIST competition.

As far as *signatures* are concerned, the BLISS scheme of Ducas et al. [5] has proved a particularly interesting target. BLISS is possibly the most efficient lattice-based signature scheme so far, comes with a public implementation [6], and has seen deployment in the wild as part of the open source VPN software suite strongSwan [11]. With reasonably small key sizes, fast signatures and strong theoretical security guarantees, it appeared as an especially attractive candidate for practical postquantum signing — until side-channel analysts started poking at it.

It turns out that the various technical ideas that form the basis of BLISS's efficiency and security all present new challenges with respect to secure implementations. The reliance on discrete Gaussian sampling,

***Corresponding Author: Mehdi Tibouchi:** 3–9–11 Midori-cho, Musashino-shi, Tokyo 180–8585, Japan; Email: mehdi.tibouchi.br@hco.ntt.co.jp
Alexandre Wallet: 3–9–11 Midori-cho, Musashino-shi, Tokyo 180–8585, Japan; Email: alexandre.wallet.th@hco.ntt.co.jp

in particular, has led to multiple side-channel attacks [4, 10], and so has the rejection sampling in signature generation [3, 7], which is essential for security. The latter has recently been broken even in the presence of certain heuristic countermeasures [2].

One other original aspect of BLISS that hasn't been specifically targeted in previous attacks is the use of *bimodal* Gaussians,[1] which manifests itself in the signature algorithm through a random sign flip. In this paper, we analyze to what extent the leakage of that sign flip also leads to a recovery attack on BLISS.

Like most of the attacks cited above, this attack simply takes advantage of the fact that neither the original implementation of BLISS nor the one included in strongSwan are *constant-time*: these implementations have conditional branches and memory accesses that depend on sensitive variables. In particular, the bit indicating whether or not the sign flip should be carried out during signature generation is sensitive, and these implementations branch on the value of that bit. Doing so reveals the bit in the timing leakage model (which can be concretely achieved through various attack techniques such as instruction cache timing attacks, simple power analysis, branch tracing, etc.), and hence lets us carry our attack. It leads to the full exposure of the signing key given that single bit of leakage for a relatively limited number of signatures: on the order of a few tens of thousands (which is comparable to [2] and [3], despite the fact that those attacks rely on a significantly larger amount of leakage per signature).

The actual recovery of the secret key $\mathbf{s}$ is carried out using parametric inference techniques, and specifically a maximum likelihood estimation. One challenge, however, is that the likelihood does not have a maximum if the parameter $\mathbf{s}$ is regarded as unconstrained. To find an actual $\mathbf{s}$, we restrict the parameter space to a compact submanifold of $\mathbb{R}^n$, and specifically a sphere, which we can do since the Euclidean norm of $\mathbf{s}$ is a public constant. The likelihood function is a smooth, strictly log-concave function on that manifold, and hence the maximum likelihood estimator $\hat{\mathbf{s}}$ is well-defined. Furthermore, the rate of convergence of $\hat{\mathbf{s}}$ is given by the Fisher information metric on the manifold, which we can compute to derive an estimate on the number of signatures required for recovery.

The countermeasure against this attack is well-known: simply carry out the sign flip in constant-time, using for example constant-time swaps. This is done correctly in the very recent constant-time implementation of BLISS described in [2].

2 Preliminaries

2.1 Notation

Vectors are written in bold letters. The zero vector is denoted by $\mathbf{0}$, and the identity matrix of dimension is $\mathbf{I}_n$. The transpose of a matrix $\mathbf{A}$ is $\mathbf{A}^\top$. We let $\|\mathbf{v}\|$, resp. $\|\mathbf{v}\|_\infty$ be the Euclidean, resp. ℓ_∞ norm of a vector $\mathbf{v}$. The set of n-dimensional vectors with entries in $\{0, 1\}$ and hamming weight κ is denoted by $\mathbb{B}_\kappa^n$. For any set S contained in the domain of a function f, we write $f(S) = \sum_{x \in S} f(s)$.

For a random variable $\mathbf{X}$, we let $\mathbb{E}_{\mathbf{X}}[\mathbf{X}]$ be its expectation and $\mathbb{V}[\mathbf{X}]$ its variance. If $\mathbf{X}$ follows a distribution μ, we write $\mathbf{X} \sim \mu$. A sample $\mathbf{x}$ for a random variable $\mathbf{X} \sim \mu$ is denoted by $\mathbf{x} \leftarrow \mu$. If μ is a distribution over a set S and $f : S \to \mathbb{R}$ is any function, we let $\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] = \sum_{\mathbf{x} \in S} f(\mathbf{x})\mu(\mathbf{x})$.

The gradient of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is denoted by $\mathbf{grad}\, f$.

1 Technically, the fact that the attacks of [2, 3] on the hyperbolic cosine part of the rejection sampling exist is also related the bimodal shape of the distribution of $\mathbf{z}$ (since that hyperbolic cosine rejection would not be needed otherwise), but this is a rather indirect relationship.

2.2 Mathematical background

Gaussian measures.
The (spherical) Gaussian function centered at $\mathbf{c} \in \mathbb{R}^n$ and "standard deviation" $\sigma > 0$ is defined for all $\mathbf{x} \in \mathbb{R}^n$ as $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\|\mathbf{x}-\mathbf{c}\|^2/(2\sigma^2)\right)$. When the center is $\mathbf{0}$, it is usually omitted. The discrete Gaussian over the lattice coset $\mathbf{c}+\mathbb{Z}^n$ with "standard deviation" σ is then

$$\forall\, \mathbf{x} \in \mathbf{c}+\mathbb{Z}^n,\ D^n_{\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\mathbf{c}+\mathbb{Z}^n)}.$$

The continuous Gaussian function restricted to $\mathbb{Z}^n$ can have a different behavior than its discrete counterpart for small values of the standard deviation σ. In our work, σ is always large enough so that the two distribution have essentially identical behavior (it exceeds the *smoothing parameter* of $\mathbb{Z}$), so that we do not need to make distinction.

Gaussian random variables are known to satisfy strong tail bounds. The following formulation is borrowed from [3, Lemma 4.4].

Theorem 2.1. *Let* $\mathbf{x}$ *be a Gaussian vector over* $\mathbb{R}^n$ *with standard deviation* σ. *For any* $t > 0$, *we have* $\Pr_{\mathbf{x}}[\|\mathbf{x}\|_\infty > t] \le 2n\exp(-\frac{t^2}{2\sigma^2})$.

Parametric statistics.
Let $\mu_{\mathbf{s}}$ be a family of probability distributions parametrized by an element $\mathbf{s}$ of some parameter space $\mathcal{S}$. In our context, the mass function $\mu_{\mathbf{s}}$ will typically be the distribution of the output of a cryptographic algorithm depending on a secret data $\mathbf{s}$.

The probability $\mu_{\mathbf{s}}(\mathbf{x})$ that a random variable $\mathbf{X} \sim \mu_{\mathbf{s}}$ takes the value $\mathbf{x}$ can be seen as a function of $\mathbf{s}$, called the *likelihood function* associated with the sample $\mathbf{x}$. It is customary to consider instead its logarithm, the *log-likelihood*, denoted by $\hat{\ell}_{\mathbf{x}}$:

$$\hat{\ell}_{\mathbf{x}}(\mathbf{s}) = \log \mu_{\mathbf{s}}(\mathbf{x}).$$

More generally, if $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are samples from independent, identically distributed random variables $\mathbf{X}_1, \ldots, \mathbf{X}_m \sim \mu_{\mathbf{s}}$, the probability $\Pr[\forall i,\ \mathbf{X}_i = \mathbf{x}_i]$ is simply the product $\prod_i \mu_{\mathbf{s}}(\mathbf{x}_i)$. We can thus define the log-likelihood function associated with these m samples as:

$$\hat{\ell}_{\mathbf{x}_1,\ldots,\mathbf{x}_m}(\mathbf{s}) = \sum_{i=1}^{m} \log \mu_{\mathbf{s}}(\mathbf{x}_i).$$

When the context is clear, we may omit the corresponding indices.

The log-likelihood can also be seen as a *random variable* on the same space of outcomes as the random variable $\mathbf{X}$ (resp. the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_m$). In that case, we denote it without a hat, as $\ell_{\mathbf{X}}$ (resp. $\ell_{\mathbf{X}_1,\ldots,\mathbf{X}_m}$).

Given samples $\mathbf{x}_1, \ldots, \mathbf{x}_m$ from i.i.d. random variables $\mathbf{X}_1, \ldots, \mathbf{X}_m \sim \mu_{\mathbf{s}}$, a *maximum likelihood* is a point $\mathbf{s}$ on the parameter space where the (log-)likelihood function reaches its maximum (if it exists). A *maximum likelihood estimator* associated with $\mathbf{X}_1, \ldots, \mathbf{X}_m \sim \mu_{\mathbf{s}}$ is a statistic $\hat{\mathbf{s}}_m$ depending on those random variables, taking values in the parameter space $\mathcal{S}$, such that for all samples $\mathbf{x}_1, \ldots, \mathbf{x}_m$, $\hat{\mathbf{s}}_m(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is a maximum likelihood:

$$\hat{\mathbf{s}}_m(\mathbf{x}_1, \ldots, \mathbf{x}_m) = \operatorname{argmax}_{\mathbf{s}} \hat{\ell}_{\mathbf{x}_1,\ldots,\mathbf{x}_m}(\mathbf{s}).$$

Note that a maximum likelihood estimator is in particular a random variable itself, with values in $\mathcal{S}$.

In case the parameter space $\mathcal{S}$ is an open subset of $\mathbb{R}^n$, and the log-likelihood function satisfies suitable regularity conditions, we can define the *Fisher information matrix* as the negative expectation of the Hessian matrix of the log-likelihood associated with a single $\mathbf{X} \sim \mu_{\mathbf{s}}$:

$$I(\mathbf{s}) = -\mathbb{E}_{\mathbf{X}}\left[\frac{\partial^2}{\partial \mathbf{s}_i \partial \mathbf{s}_j} \ell_{\mathbf{X}}(\mathbf{s})\right].$$

Then, again under suitable regularity conditions, $I(\mathbf{s})$ is symmetric positive definite, and the following standard theorem holds.

Theorem 2.2 ([8, Th. 6.4.1]). *Consider a sequence $(\mathbf{X}_i)_{i\geq 1}$ of i.i.d. random variables $\mathbf{X}_i \sim \mu_{\mathbf{s}}$. There exists a sequence $(\hat{\mathbf{s}}_m)_{m\geq 1}$ such that $\hat{\mathbf{s}}_m$ is a maximum likelihood estimator associated with $\mathbf{X}_1, \ldots, \mathbf{X}_m$ for each m, and that $\hat{\mathbf{s}}_m$ converges to $\mathbf{s}$ almost surely. Moreover, for any such sequence, the random variable $\sqrt{m}(\hat{\mathbf{s}}_m - \mathbf{s})$ converges in distribution to $\mathcal{N}(\mathbf{0}, I(\mathbf{s})^{-1})$.*

We refer to [8] and [9, Ch. 6] for more precise statements including all the regularity conditions. In this work, it is easy to check that the regularity conditions are always satisfied. Another interpretation of the above theorem states that maximum likelihood estimators are asymptotically efficient (in the sense that they reach the so-called Cramér-Rao bound, see again [9, Ch. 6]). In other words, they are estimators that requires the less amount of samples to estimate the real result.

One way in which our setting differs, however, is that in our case, the parameter space $\mathcal{S}$ is not an open subset of $\mathbb{R}^n$, but instead a closed submanifold—specifically a sphere. The definition of the Fisher information matrix and Theorem 2.2 generalize naturally to that setting as well. Roughly speaking, the Fisher information matrix becomes a metric tensor on the manifold (which can be seen as a positive definite matrix in the local coordinates in each tangent space). We refer to [1, Ch. 5] for a detailed discussion. In our setting, the log-likelihood function on $\mathcal{S}$ is the restriction to $\mathcal{S}$ of a smooth function defined over all of $\mathbb{R}^n$, so its gradient and Hessian are simply obtained as the projection on the tangent space (resp. the restriction to the tangent space) of the gradient and Hessian of the function on all of $\mathbb{R}^n$.

Representation of ring elements.
BLISS is described over cyclotomic rings such as $R = \mathbb{Z}[x]/(x^n + 1)$, where $n = 2^\lambda$ for some $\lambda \geq 1$. There are several known representations for elements, and we will make no distinction between them. An element $\mathbf{u} = \sum_i u_i x^i \in R$ can be seen as the integer vector $(u_0, \ldots, u_{n-1})$. This allows to identify R as the lattice $\mathbb{Z}^n$ in the ambient inner-product space $(\mathbb{R}^n, \langle\,,\,\rangle)$. Elements can also be represented by their matrix of multiplication in the basis $1, x, \ldots, x^{n-1}$. This allows to see that $\mathbf{uv}$ corresponds to the vector whose i-th coordinate is $\langle x^i\mathbf{u}, \mathbf{v}\rangle$. The adjoint $\mathbf{v}^\star$ of $\mathbf{v}$ is the ring element whose multiplication matrix is the transpose of the matrix of multiplication by $\mathbf{v}$. It is given by $\mathbf{v}^\star = v_0 - v_{n-1}x - \cdots - v_1 x^{n-1}$, and satisfies $\langle \mathbf{u}, \mathbf{vw}\rangle = \langle \mathbf{uv}^\star, \mathbf{w}\rangle$.

2.3 The BLISS scheme

BLISS [5] is a lattice-based signature scheme based on the Ring-LWE assumption. The signature generation (Algorithm 1) involves rejection sampling, which is fundamental toward security. Indeed, it is used to guarantee that the distribution of the signatures does not depend on that of the secret key $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$. More precisely, a candidate signature $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2)$ for a message μ is kept only with probability

$$\frac{1}{M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\langle \mathbf{Z}, \mathbf{Sc}\rangle/\sigma^2\right)}. \tag{1}$$

The above probability must be computed at each try to perform the rejection, providing leakage on $\mathbf{S}$ if not done in constant-time.

Algorithm 1: Sign(μ, pk = ($\mathbf{v}_1$, $q-2$), sk = $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$)

1: $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D_\sigma^n$;
2: $\mathbf{u} \leftarrow \zeta \cdot \mathbf{v}_1 \cdot \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$;
3: $\mathbf{c} \leftarrow \mathcal{H}(\lceil \mathbf{u} \rfloor_d \bmod p, \mu)$;
4: Choose a random bit b;
5: $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$, and $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$;
6: **continue** with probability $1/(M \exp(-\|\mathbf{Sc}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{Z}, \mathbf{Sc} \rangle / \sigma^2)$; else **restart**;
7: $\mathbf{z}_2^\dagger \leftarrow (\lceil \mathbf{u} \rfloor_d - \lceil \mathbf{u} - \mathbf{z}_2 \rfloor_d) \bmod p$;
8: **return** $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$;

During the key generation algorithm, $\mathbf{s}_1$ and $\mathbf{s}_2$ are sampled uniformly among the set of n-dimensional vectors with $\lceil \delta_1 n \rfloor$ entries in $\{\pm 1\}$ and $\lceil \delta_2 n \rfloor$ entries in $\{\pm 2\}$ for some small densities $\delta_1, \delta_2 \in (0, 1)$. The rest of the entries are zero. The first component of the public key is defined as $\mathbf{v}_1 = (2\mathbf{s}_2 + 1)\mathbf{s}_1^{-1} \bmod q$. The element ζ satisfies $\zeta \cdot (q-2) = 1 \bmod 2q$. The authors of [5] model the hash function $\mathcal{H}$ of Step 3 as a random oracle with output in $\mathbb{B}_\kappa^n$, for some small parameter[2] $\kappa > 0$. After rejection, it can be shown that, without additional information, the distribution of $\mathbf{z}_1$ is indistinguishable from that of a bimodal Gaussian distribution with standard deviation parameter σ.

3 One bit is all it takes

Observe that to break the scheme, it is enough to recover the $\mathbf{s}_1$ part of the signature, as $\mathbf{s}_2$ can be directly obtained from the knowledge of $\mathbf{s}_1$ and the public key. Hence, we write $\mathbf{s} = \mathbf{s}_1$ and focus on this component for the rest of this section. Accordingly, we also write $\mathbf{z} = \mathbf{z}_1$. BLISS key-generation and signature consider only spherical Gaussians distribution for their sampling. In particular, all the coordinates of the generated samples are independent. This allows to rewrite the rejection probability in term of any target subset of coordinates for $\mathbf{S}$ and $\mathbf{Z}$, so we can focus on $\mathbf{s}$ and $\mathbf{z}$ by considering accordingly their rejection probability.

Then, our attack boils down to the fact that knowing b allows to express the log-likelihood functions for m samples of the distribution of the output of Algorithm 1. We start by calculating it explicitly and deduce that, on a n-dimensional sphere of radius $\|\mathbf{s}\| = \lceil \delta_1 n \rfloor + 4\lceil \delta_2 n \rfloor$ (which is known), it admits a unique maximum likelihood estimator.

This calculation also enables us to express the Fisher information matrix $I(\mathbf{s})$. We then assess the number of needed traces to recover (almost all of) $\mathbf{s}$ with high probability, relying first on a heuristic analysis of the behavior of $I(\mathbf{s})$ for the sake of clarity. In Appendix A, we give more rigorous justifications for our heuristic arguments. The final result is then obtained combining Theorem 2.2 and the Gaussian tail bound. Algorithmic and practical aspects of the attack are dealt with in Section 4.

3.1 Existence of the maximum likelihood estimator

Let $\mathcal{A}$ be the distribution of $(b, \mathbf{z}, \mathbf{c})$ outputted by Algorithm 1, and consider $(b, \mathbf{z}, \mathbf{c}) \leftarrow \mathcal{A}$ as well. From Step 5 it is clear that before rejection sampling, $\mathbf{z}$ is distributed as

$$D^n_{\sigma,(-1)^b\mathbf{sc}}(\mathbf{z}) = \frac{1}{\rho_\sigma(\mathbb{Z}^n)} \cdot \exp\left(-\frac{\|\mathbf{z} + (-1)^b \mathbf{sc}\|^2}{2\sigma^2}\right).$$

2 For example, the parameter set for BLISS–I where $n = 512$ suggests $\delta_1 = 0.3$, $\delta_2 = 0$ and $\kappa = 23$.

Combining with the rejection probability (1), we readily see that the probability to get a specific output $(b, \mathbf{z}, \mathbf{c})$ is

$$\mu_{\mathbf{s}}(b, \mathbf{z}, \mathbf{c}) = \frac{2\exp\left(\frac{-\|\mathbf{z}\|^2}{2\sigma^2}\right)}{\rho_\sigma(\mathbb{Z}^n)} \cdot \frac{\exp\left(\frac{(-1)^b\langle \mathbf{z}, \mathbf{sc}\rangle}{\sigma^2}\right)}{\exp\left(\frac{\langle \mathbf{z}, \mathbf{sc}\rangle}{\sigma^2}\right) + \exp\left(-\frac{\langle \mathbf{z}, \mathbf{sc}\rangle}{\sigma^2}\right)}. \tag{2}$$

The log-likelihood function associated to one sample $(b, \mathbf{z}, \mathbf{c})$ is then $\hat{\ell}_{(b,\mathbf{z},\mathbf{c})}(\mathbf{s}) = K - \log\left(1+\exp(-2\langle \mathbf{z}, \mathbf{sc}\rangle/\sigma^2)\right)$, where K is constant with respect to $\mathbf{s}$. Similarly, the log-likelihood function associated to m samples $(b^{(i)}, \mathbf{z}^{(i)}, \mathbf{c}^{(i)}) \leftarrow \mathcal{A}$ is:

$$\hat{\ell}_{(b^{(i)},\mathbf{z}^{(i)},\mathbf{c}^{(i)})_i}(\mathbf{s}) := K' - \sum_{i=1}^{m} \log\left(1 + \exp\left(-\frac{2}{\sigma^2}\langle (-1)^b \mathbf{z}^{(i)}(\mathbf{c}^{(i)})^\star, \mathbf{s}\rangle\right)\right).$$

where again, K' is a constant independent of $\mathbf{s}$. As $\|\mathbf{s}\|$ is known from the description of the scheme, a natural set where to maximize $\hat{\ell}_{(b^{(i)},\mathbf{z}^{(i)},\mathbf{c}^{(i)})_i}$ is the sphere $\mathcal{S}$ of radius $\|\mathbf{s}\|$ in $\mathbb{R}^n$. Since $\hat{\ell}_{(b^{(i)},\mathbf{z}^{(i)},\mathbf{c}^{(i)})_i}$ is clearly continuous on this compact set, it is bounded and reaches its maximum on $\mathcal{S}$. We can see moreover that the point at which this maximum is necessarily unique, and therefore determines a unique, well-defined maximum likelihood estimator $\hat{\mathbf{s}}$.

Indeed, let $\varphi(t) = -\log\left(1 + \exp\left(-\frac{2t}{\sigma^2}\right)\right)$. Straightforward computations give:

$$\varphi'(t) = \frac{-2}{\sigma^2(1+\exp(2t/\sigma^2))} \quad \text{and} \quad \varphi''(t) = \frac{-1}{\sigma^4\cosh(t/\sigma^2)^2}.$$

This implies that for any $\mathbf{w} \in \mathbb{R}^n$, $\varphi(\langle \mathbf{w}, \mathbf{s}\rangle)$ is a *strictly concave* function of $\mathbf{s}$, and therefore so is $\hat{\ell}_{(b^{(i)},\mathbf{z}^{(i)},\mathbf{c}^{(i)})_i}$. It follows that the maximum of $\hat{\ell}$ is reached at a unique point of $\mathcal{S}$ as stated.

3.2 Determining the number of required traces

For any fixed $\mathbf{w} \in \mathbb{R}^n$, we readily compute $\frac{\partial^2}{\partial \mathbf{s}_j \partial \mathbf{s}_k}\hat{\ell} = \varphi''(\langle \mathbf{w}, \mathbf{s}\rangle) \cdot \mathbf{w}_j\mathbf{w}_k$ for any $j, k \in [n]$, where $\mathbf{w}_j$ is the j-th coordinate of $\mathbf{w}$. Let now $(b, \mathbf{z}, \mathbf{c}) \sim \mathcal{A}$ and define $\mathbf{w} = (-1)^b\mathbf{z}\mathbf{c}^\star$. We can express the Fisher information matrix as

$$I(\mathbf{s}) = -\mathbb{E}_{\mathbf{w}}\left[\left(\varphi''(\langle \mathbf{w}, \mathbf{s}\rangle) \cdot \mathbf{w}_j\mathbf{w}_k\right)_{j,k}\right],$$

where $\mathbf{w}$ follows a distribution to be fully described later. Nevertheless, it is checked from its definition that all coordinates of $\mathbf{w}$ are mutually independent variables. Now let $\overline{\mathbf{w}}$ be the vector expected for $\mathbf{w}$. Using the independence of the coordinates of $\mathbf{w}$, we have

$$I(\mathbf{s})_{j,k} = \begin{cases} \mathbb{E}_{\mathbf{w}}\left[-\varphi''(\langle \mathbf{w}, \mathbf{s}\rangle)\right] \cdot \overline{\mathbf{w}}_j\overline{\mathbf{w}}_k \text{ if } j \neq k, \\ \mathbb{E}_{\mathbf{w}}\left[-\varphi''(\langle \mathbf{w}, \mathbf{s}\rangle)\right] \cdot (\overline{\mathbf{w}}_j^2 + \mathbb{V}[\mathbf{w}_j]) \text{ if } j = k. \end{cases}$$

Since $(-1)^b\mathbf{z} \sim D^n_{\sigma,\mathbf{sc}}$ and since $\mathbf{c}^\star$ has exactly κ non zero entries, we see that $\mathbb{V}[\mathbf{w}_j] = \kappa\sigma^2$ for all j. We deduce that

$$I(\mathbf{s}) = \mathbb{E}_{\mathbf{w}}\left[\cosh\left(\frac{\langle \mathbf{w}, \mathbf{s}\rangle}{\sigma^2}\right)^{-2}\right] \cdot \frac{1}{\sigma^4}\left(\kappa\sigma^2\mathbf{I}_n + \overline{\mathbf{w}} \cdot \overline{\mathbf{w}}^\top\right).$$

Heuristic analysis of $I(\mathbf{s})$

Our goal is now to argue that $I(\mathbf{s})^{-1}$ essentially behaves like a scalar matrix $(\sigma^2/\kappa)\mathbf{I}_n$. In order to do this, we need to know the distribution of $\mathbf{w}$. We show in Appendix A that it is close to $\mathcal{N}(\kappa \cdot \mathbf{s}, \sigma\sqrt{\kappa})$. For the rest of the current section, we therefore assume that $\mathbf{w}$ follows that particular normal law.

Under this assumption, we can write $\langle \mathbf{w}, \mathbf{s}\rangle = \kappa\|\mathbf{s}\|^2 + \langle \mathbf{h}, \mathbf{s}\rangle$. As the non-zero coordinates of $\mathbf{s}$ are sampled uniformly in centered sets, we expect that is has essentially the same number of positive and negative coordinates. More precisely, potentially "problematic" $\mathbf{s}$'s (with an imbalanced number of positive or negative

coordinates) exist only in negligible proportion. As the coordinates of $\mathbf{h}$ are at most $\sigma\sqrt{\kappa}$ with overwhelming probability, we expect $|\langle \mathbf{h}, \mathbf{s}\rangle|$ to be within a small factor from $\sigma\sqrt{\kappa}$.

Combined with a look at usual BLISS parameters, this suggests that $\langle \mathbf{w}, \mathbf{s}\rangle/\sigma^2$ is likely to be close to 0 for most of $\mathbf{w}$'s and a fixed $\mathbf{s}$. We deduce that

$$I(\mathbf{s}) \approx \frac{\kappa}{\sigma^2}\left(\mathbf{I}_n + \frac{1}{\kappa\sigma^2}\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right).$$

The rank-1 matrix $\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top$ has a unique non-zero eigenvalue given by $\overline{\mathbf{w}}^\top\overline{\mathbf{w}} = \kappa^2\|\mathbf{s}\|^2$. This means that $\frac{1}{\kappa\sigma^2}\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top$ has operator norm smaller than 1, which shows that $I(\mathbf{s})$ is invertible. As σ^2 is at least an order of magnitude larger than $\kappa\|\mathbf{s}\|^2$ for BLISS parameters, this is enough to argue that $I(\mathbf{s})^{-1}$ is close to $(\sigma^2/\kappa)\mathbf{I}_n$.

Recall that $\hat{\mathbf{s}}$ is the maximum likelihood estimator associated to m samples from $\mathcal{A}$. From Theorem 2.2, we infer that for m large enough, $\hat{\mathbf{s}} - \mathbf{s}$ is a centered Gaussian vector of covariance $(\sigma^2/\kappa m)\mathbf{I}_n$. We do not consider the speed of convergence and assume that the result is "valid enough" for the ranges of m that we are interested in. Then, Theorem 2.1 states that $\|\hat{\mathbf{s}}-\mathbf{s}\|_\infty \le 1/2$ except with probability less than $2n\exp(-\kappa m/8\sigma^2)$. Therefore, taking $m \ge 16\log(2n)\sigma^2/\kappa$ gives that $\|\hat{\mathbf{s}} - \mathbf{s}\|_\infty \le 1/2$ except with probability at most $1/2n$. As $\hat{\mathbf{s}}$ is essentially an optimal estimator, we remark that it should not be possible to improve the number of required samples more than marginally.

4 Implementation of the attack

In this section, we describe how to mount the attack in practice from an algorithmic standpoint given the required leakage, and provide some experimental results. We also discuss to what extent the publicly available implementations of BLISS are vulnerable to this attack. The code for the attack can be found at https://github.com/awallet/OneBitBliss.

4.1 Algorithmic considerations

Given m signatures together with the corresponding leakage bits $\left(b^{(i)}, \mathbf{z}^{(i)}, \mathbf{c}^{(i)}\right)$, we can form the vectors $\mathbf{w}^{(i)} = (-1)^b\mathbf{z}^{(i)}(\mathbf{c}^{(i)})^\star$. As seen in Section 3, the log-likelihood function $\hat{\ell}$ and its gradient are expressed as follows:

$$\hat{\ell}(\mathbf{s}) = \sum_{i=1}^{m}\varphi\left(\langle \mathbf{w}^{(i)}, \mathbf{s}\rangle\right) \quad \text{and} \quad \mathbf{grad}\,\hat{\ell}(\mathbf{s}) = \sum_{i=1}^{m}\varphi'\left(\langle \mathbf{w}^{(i)}, \mathbf{s}\rangle\right)\mathbf{w}^{(i)}$$

when they are seen as functions on $\mathbb{R}^n$. It is therefore straightforward in principle to evaluate them numerically given the available data, and hence use an algorithmic such as gradient descent (or rather, gradient *ascent*) to search for the corresponding maximum likelihood estimator $\hat{\mathbf{s}}$.

Recall however that we are doing maximization *on the sphere* centered at $\mathbf{0}$ and of radius $\|\mathbf{s}\|$. Hence, the gradient ascent should also be carried out on the sphere, along geodesic paths. Concretely, each iteration of the gradient ascent is carried out as follows.

(i) Starting from a point $\mathbf{v}$ on the sphere, we compute $\hat{\ell}(\mathbf{v})$ as well as the gradient $\mathbf{g} = \mathbf{grad}\,\hat{\ell}(\mathbf{v})$ at $\mathbf{v}$ as above. We then compute the projection $\mathbf{h}$ of $\mathbf{g}$ on the tangent plane $\mathbf{v}^\perp$ at $\mathbf{v}$ as $\mathbf{h} = \mathbf{g} - \frac{\langle g, v\rangle}{\|s\|^2}\mathbf{v}$.

(ii) The vector $\mathbf{h}$ gives the direction of steepest ascent for $\hat{\ell}$ on the sphere, and the geodesic through $\mathbf{v}$ in the direction of $\mathbf{h}$ is parametrized by $\mathbf{v}'(\theta) = \cos\theta\cdot\mathbf{v} + \sin\theta\cdot\mathbf{h}$. We take a step along that geodesic using backtracking line search as described in the next steps. Note that the first order Taylor expansion around $\theta = 0$ gives:

$$\begin{aligned}\hat{\ell}\left(\mathbf{v}'(\theta)\right) &= \hat{\ell}\left(\cos\theta\cdot\mathbf{v} + \sin\theta\cdot\mathbf{h}\right) = \hat{\ell}\left(\mathbf{v} + \theta\mathbf{h} + o(\theta)\right)\\ &= \hat{\ell}(\mathbf{v}) + \theta\langle \mathbf{g}, \mathbf{h}\rangle + o(\theta) = \hat{\ell}(\mathbf{v}) + \|\mathbf{h}\|^2\cdot\theta + o(\theta).\end{aligned}$$

(iii) Initialize θ to θ_0 and compute $\ell_0 = \hat{\ell}(\mathbf{v}'(\theta_0))$. If the "Armijo condition" $\ell_0 \geq \hat{\ell}(\mathbf{v}) + \|\mathbf{h}\|^2 \cdot \theta/2$ is satisfied (which, by the above, must happen for θ small enough), keep $\mathbf{v}' = \mathbf{v}'(\theta_0)$ as the result for this iteration of the gradient ascent. Otherwise, update $\theta \leftarrow \nu\theta$ for some constant $\nu < 1$ and try again.

This gradient ascent procedure is then repeated for a certain number of iterations or until $\|\mathbf{h}\|$ becomes sufficiently small. It is standard that the concavity of $\hat{\ell}$ ensures the convergence to the maximum.

We also need to specify an initial vector $\mathbf{v}$ to initialize the gradient ascent. For this, we use the fact that the maximum likelihood $\hat{\mathbf{s}}$ for a single sample $\mathbf{w}$ is exactly $\lambda\mathbf{w}$, where λ is the positive normalization factor needed to obtain a point on the sphere. This is because φ is an increasing function, and hence the maximum likelihood is obtained by maximizing the inner product $\langle \mathbf{w}, \hat{\mathbf{s}} \rangle$. Thus, as a first approximation for the maximum likelihood of m samples $\mathbf{w}^{(i)}$, we simply pick the vector $\mathbf{v} = \lambda \sum_i \mathbf{w}^{(i)}$, where λ is again chosen so that $\mathbf{v}$ in on the sphere. We find that this choice gives good results (although in principle, convergence could be achieved from any starting point on the sphere anyway).

4.2 Experimental results

We implemented the algorithm of the previous section and ran it on samples generated by the original implementation of BLISS, together with the leakage of the bit b. The source code of the attack is attached to this submission as supplementary material.

For each of the BLISS parameters BLISS–I to BLISS–IV, we generated 100 fresh key pairs, together with corresponding samples in batches of 20,000, and counted for each key pair how many batches where needed for the attack of the previous section to succeed and fully recover the secret key. The lower quartile (LQ), median, and upper quartile (UQ) numbers of signatures m are all reported in Table 1. We also counted how many batches allowed to recover $n' = 504$ out of the 512 secret key coefficients, since such a recovery can then be combined with a simple meet-in-the-middle attack of low complexity (namely $n \cdot \binom{n/2}{n'/2} \approx 2^{36}$ time and $\binom{n/2}{n'/2} \approx 2^{27}$ space) to deduce the exact secret key.

Table 1: Results of our experiments.

BLISS–	I	II	III	IV
Theoretical m for success: $16\log(2n)\sigma^2/\kappa$	223,000	55,000	231,000	209,000
Experimental m for full recovery (LQ)	120,000	60,000	160,000	170,000
Experimental m for full recovery (median)	130,000	70,000	180,000	200,000
Experimental m for full recovery (UQ)	150,000	80,000	200,000	230,000
Experimental m for n'/n recovery (LQ)	70,000	40,000	90,000	110,000
Experimental m for n'/n recovery (median)	70,000	40,000	100,000	110,000
Experimental m for n'/n recovery (UQ)	80,000	40,000	110,000	120,000

As we can see from the table, our analysis predicts the right order of magnitude for m, even though our gradient ascent only gives an approximation of the maximum likelihood, and despite the various heuristics involved (in particular, ignoring the fact that the maximum likelihood is only unbiased *in the limit* and not necessarily for a bounded number of samples). In addition, combining our approach with a meet-in-the-middle attack does significantly reduce the number of required signatures for successful recovery.

Finally, we note that these results all consider the attack on the first component $\mathbf{z}_1$ of the signature. In principle, the attack on the $\mathbf{z}_2$ component should be more efficient, due to the fact that the coefficients of $\mathbf{s}_2$ (except possibly the first one) are all in $2\mathbb{Z}$, and hence obtaining an estimator $\hat{\mathbf{s}}_2$ with $\|\hat{\mathbf{s}}_2 - \mathbf{s}_2\|_\infty < 1$ (instead of $1/2$) suffices for recovery, which reduces the required number of signatures by a factor of 4. In practice, however, this is impractical, due to the fact that in actual BLISS signatures, the second component $\mathbf{z}_2$ is not

given out in full, but only in compressed form $\mathbf{z}_2^\dagger$, and the compression is lossy—it only allows to recover an approximation of the corresponding samples $\mathbf{w}$. For most parameter sets, the approximation is rather loose, and this makes the attack on the second component significantly worse than the attack on $\mathbf{z}_1$. For BLISS–IV, however, the compression is not so lossy, and it turns out that the attack on $\mathbf{z}_2$ is in fact measurably better despite the compression (with full recovery possible given 40,000 to 50,000 signatures).

```
if (random->getRandomBit()) {
  for (i=0; i<N; i++) {
    signOutput.z1[i]-=sc1[i];
    signOutput.z2[i]-=sc2[i];
  }
} else {
  for (i=0; i<N; i++) {
    signOutput.z1[i]+=sc1[i];
    signOutput.z2[i]+=sc2[i];
  }
}
```

(a) Original BLISS (`Sign.cpp`:150–159)

```
for (i = 0; i < n; i++)
{
  if (positive)
  {
    z1[i] = y1[i] + s1c[i];
    z2[i] = y2[i] + s2c[i];
  }
  else
  {
    z1[i] = y1[i] - s1c[i];
    z2[i] = y2[i] - s2c[i];
  }
}
```

(b) strongSwan (`bliss_private_key.c`:422–434)

```
void poly_sign_flip(
  polysmall_t *a,
  unsigned char sign)
{
  unsigned int i;

  for(i = 0; i < N; i++)
    a->coeffs[i] =
    CFLIP(a->coeffs[i], sign);
}
```

(c) GALACTICS (`poly.c`:80–86)

Figure 1: Existing implementations of the sign flip in BLISS.

4.3 Vulnerability of concrete implementations

The part of the signing algorithm corresponding to the sign flip is implemented as shown in Fig. 1 in the various available implementations of BLISS, namely the original one by Ducas and Lepoint [6], the implementation in strongSwan [11] and the recent constant-time implementation described in [2].

Both the original BLISS implementation (as shown in Fig. 12) and strongSwan (Fig. 14) share the same structure, and essentially branch on the bit b indicating the sign flip. On most platforms, additions and subtractions have the same running time, and thus the running time of the corresponding computation is the same regardless of the bit b. Nevertheless, these implementations do not qualify as *constant-time*, because they contain a conditional branch on the sensitive value b. This can be exploited in practice using side-channels such as a *cache timing attack* on the instruction cache (provided that the instructions corresponding to the two branches end up in different cache lines) or the *branch tracing attack* described in [7]. Branch tracing breaks both implementation equally easily; cache timing attacks should be easier to mount on strongSwan, however, since the branch is taken in every iteration of the `for` loop instead of just once in Fig. 12 (although compiler optimizations might affect this distinction in practice).

In contrast, the implementation of Fig. 11, described in [2], is unaffected by this attack, since the sign flip is carried out using a constant-time expression. More precisely, The `CFLIP(x,b)` macro essentially expands to `(x&(-!b))^((-x)&(-b))`, and hence is computed using only bitwise operations, without any conditional branches.

References

[1] Nihat Ay, Jürgen Jost, Hông Vân Lê and Lorenz Schwachhöfer, *Information Geometry*, Springer, 2017.

[2] Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Mélissa Rossi and Mehdi Tibouchi, GALACTICS: Gaussian Sampling for Lattice-Based Constant- Time Implementation of Cryptographic Signatures, Revisited, in: *ACM CCS 2019* (Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang and Jonathan Katz, eds.), pp. 2147–2164, ACM Press, November 2019.

[3] Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque and Mehdi Tibouchi, LWE Without Modular Reduction and Improved Side-Channel Attacks Against BLISS, in: *ASIACRYPT 2018, Part I* (Thomas Peyrin and Steven Galbraith, eds.), LNCS 11272, pp. 494–524, Springer, Heidelberg, December 2018.

[4] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange and Yuval Yarom, Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme, in: *CHES 2016* (Benedikt Gierlichs and Axel Y. Poschmann, eds.), LNCS 9813, pp. 323–345, Springer, Heidelberg, August 2016.

[5] Léo Ducas, Alain Durmus, Tancrède Lepoint and Vadim Lyubashevsky, Lattice Signatures and Bimodal Gaussians, in: *CRYPTO 2013, Part I* (Ran Canetti and Juan A. Garay, eds.), LNCS 8042, pp. 40–56, Springer, Heidelberg, August 2013.

[6] Léo Ducas and Tancrède Lepoint, *BLISS: Bimodal Lattice Signature Schemes*, June 2013, http://bliss.di.ens.fr/bliss-06-13-2013.zip (proof-of-concept implementation).

[7] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard and Mehdi Tibouchi, Side-Channel Attacks on BLISS Lattice-Based Signatures: Exploiting Branch Tracing against strongSwan and Electromagnetic Emanations in Microcontrollers, in: *ACM CCS 2017* (Bhavani M. Thuraisingham, David Evans, Tal Malkin and Dongyan Xu, eds.), pp. 1857–1874, ACM Press, October / November 2017.

[8] Robert V. Hogg, Joseph W. McKean and Allen T. Craig, *Introduction to Mathematical Satistics (8th edition)*, Pearson, 2018.

[9] Erich L. Lehmann and George Casella, *Theory of Point Estimation*, Springer, 1998.

[10] Peter Pessl, Leon Groot Bruinderink and Yuval Yarom, To BLISS-B or not to be: Attacking strongSwan's Implementation of Post-Quantum Signatures, in: *ACM CCS 2017* (Bhavani M. Thuraisingham, David Evans, Tal Malkin and Dongyan Xu, eds.), pp. 1843–1855, ACM Press, October / November 2017.

[11] Andreas Steffen et al., *strongSwan: the Open Source IPsec-based VPN Solution (version 5.5.2)*, March 2017.

A Analyzing the distribution of w

Recall that we defined the random variable $\mathbf{w} = (-1)^b \mathbf{z}\mathbf{c}^\star$ from the random vector $(b, \mathbf{z}, \mathbf{c}) \leftarrow \mathcal{A}$, where $\mathcal{A}$ is the distribution of the output of Algorithm 1. If there was no rejection sampling, one could read the distribution of $\mathbf{y}$ from Step 5, as $\mathbf{y} = (-1)^b \mathbf{z} + \mathbf{s}\mathbf{c}$. Since $\mathbf{y}$ is large compared to $\mathbf{s}\mathbf{c}$, the rejection step makes $\mathbf{y}$ a centered Gaussian with covariance σ^2, so that $\mathbf{y}\mathbf{c}^\star$ should have covariance $\sigma^2\kappa$. This suggests that $\mathbf{w}$ should be centered at $\mathbf{s}\mathbf{c}\mathbf{c}^\star$, with the same covariance. We make this heuristic argument rigorous in the following.

From Equation (2), we can write

$$\begin{aligned}
\mathbb{E}_{b,\mathbf{z}}\left[(-1)^b \mathbf{z} \mid \mathbf{c}\right] &= \frac{1}{\rho_\sigma(\mathbb{Z}^n)} \sum_{\mathbf{z}\in\mathbb{Z}^n} \mathbf{z} \cdot \mu_{\mathbf{s}}(0, \mathbf{z}, \mathbf{c}) - \frac{1}{\rho_\sigma(\mathbb{Z}^n)} \sum_{\mathbf{z}\in\mathbb{Z}^n} \mathbf{z} \cdot \mu_{\mathbf{s}}(1, \mathbf{z}, \mathbf{c}) \\
&= \frac{1}{\rho_\sigma(\mathbb{Z}^n)} \sum_{\mathbf{z}\in\mathbb{Z}^n} \mathbf{z} \cdot \exp\left(-\frac{\|\mathbf{z}\|^2}{2\sigma^2}\right) \cdot \frac{\exp(\frac{\langle \mathbf{z}, \mathbf{s}\mathbf{c}\rangle}{\sigma^2}) - \exp(-\frac{\langle \mathbf{z}, \mathbf{s}\mathbf{c}\rangle}{\sigma^2})}{\exp(\frac{\langle \mathbf{z}, \mathbf{s}\mathbf{c}\rangle}{\sigma^2}) + \exp(-\frac{\langle \mathbf{z}, \mathbf{s}\mathbf{c}\rangle}{\sigma^2})} \\
&= \mathbb{E}_{\mathbf{z}}\left[\mathbf{z} \tanh\left(\frac{\langle \mathbf{z}, \mathbf{s}\mathbf{c}\rangle}{\sigma^2}\right) \mid \mathbf{c}\right],
\end{aligned}$$

where $\mathbf{z} \leftarrow D^n_{\sigma,\mathbb{Z}}$ on the last line.

We now claim that this expected vector is essentially $\mathbf{s}\mathbf{c}$. For the sake of clarity, we consider that $\mathbf{z}$ follows a normal law of standard deviation σ. Using the change of variables $\mathbf{z} = \sigma\mathbf{u}$, we can write

$$\mathbb{E}_{b,\mathbf{z}}\left[(-1)^b \mathbf{z} \mid \mathbf{c}\right] = \frac{\sigma}{(\sqrt{2\pi})^n} \int_{\mathbb{R}^n} \mathbf{u} \cdot \exp\left(-\frac{\|\mathbf{u}\|^2}{2}\right) \tanh(\langle \sigma^{-1}\mathbf{u}, \mathbf{s}\mathbf{c}\rangle) d\mathbf{u}. \tag{3}$$

Take any orthonormal basis of $\mathbb{R}^n$ whose first vector is $\mathbf{v} = \mathbf{s}\mathbf{c}/\|\mathbf{s}\mathbf{c}\|$, and call $\mathbb{E}_i$ the i-th coordinate of $\mathbb{E}_{b,\mathbf{z}}[(-1)^b \mathbf{z} \mid \mathbf{c}]$ in this basis. Fubini's theorem gives

$$\begin{aligned}
\mathbb{E}_1 &= \frac{\sigma}{(\sqrt{2\pi})^n} \int_{\mathbb{R}} u \cdot \exp\left(\frac{-u^2}{2}\right) \tanh(u\frac{\|\mathbf{s}\mathbf{c}\|}{\sigma}) du \cdot \left(\int_{\mathbb{R}} \exp(\frac{-t^2}{2}) dt\right)^{n-1} \\
&= \frac{\|\mathbf{s}\mathbf{c}\|}{\sqrt{2\pi}} \int_{\mathbb{R}} \frac{\exp(\frac{-u^2}{2})}{\cosh(u\frac{\|\mathbf{s}\mathbf{c}\|}{\sigma})^2} du,
\end{aligned}$$

where the second line uses an integration by parts. Letting $\varepsilon = \frac{\|\mathbf{s}\mathbf{c}\|}{\sigma}$ and using that $1 \le \cosh(t) \le \exp(t^2/2)$ for all t, a straightforward computation shows that $\mathbb{E}_1 \in \|\mathbf{s}\mathbf{c}\| \cdot [(1 + 2\varepsilon^2)^{-1}, 1]$.

Fubini's theorem also allows to see that each other coordinates involves a factor $\int_{\mathbb{R}} \tanh(u\|\mathbf{s}\mathbf{c}\|/\sigma) \exp(-u^2/2) du$. Observe that the function to be integrated in this factor is odd, so that $\mathbb{E}_i = 0$ for all $i \ge 2$. For BLISS parameters, σ is an order of magnitude larger than $\|\mathbf{s}\mathbf{c}\|$; in other words, ϵ is small. This gives the claim.

For a fixed $\mathbf{s}$, we deduce that $\mathbb{E}_{\mathbf{w}}[\mathbf{w}] = \mathbf{s}\mathbb{E}_{\mathbf{c}}[\mathbf{c}\mathbf{c}^\star]$. Our next goal is to show that the random vector $\mathbf{c}\mathbf{c}^\star$ has a center close to $(\kappa, 0, \ldots, 0) \in \mathbb{R}^n$. We observe immediately that $(\mathbf{c}\mathbf{c}^\star)_1 = \|\mathbf{c}\|^2 = \kappa$. For a fixed $\mathbf{c} \in \mathbb{B}^n_\kappa$, let χ the indicator function of its support. By standard algebraic manipulations, we find in general

$$(\mathbf{c}\mathbf{c}^\star)_{i+1} = \langle x^i \mathbf{c}, \mathbf{c}\rangle = -\sum_{k=0}^{i-1} \chi(n + k - i)\chi(k) + \sum_{k=i}^{n-1} \chi(k - i)\chi(k).$$

Assuming that the coordinates of $\mathbf{c}$ are mutually independent[3], taking the expectation for $i \ge 1$ yields $\mathbb{E}_{\mathbf{c}}[\mathbf{c}\mathbf{c}^\star]_{i+1} = -i\frac{\kappa^2}{n^2} + (n - i)\frac{\kappa^2}{n^2} = \frac{\kappa^2}{n^2}(n - 2i)$. This shows that $\|\mathbb{E}_{\mathbf{c}}[\mathbf{c}\mathbf{c}^\star] - (\kappa, 0, \ldots, 0)\|_\infty \le \kappa^2/n$, which is practically small for any proposed parameters for BLISS. This confirms that the center of $\mathbf{w}$ is close to $\kappa\mathbf{s}$.

Finally, we show that the covariance matrix of $\mathbf{w}$ is essentially $\sigma^2\kappa\,\mathbf{I}_n$. We readily see that the (i, j)-th entry in the covariance matrix of $(-1)^b\mathbf{z}$ is

$$\mathbb{V}_{ij} := \mathbb{E}_{b,\mathbf{z}_i}[\mathbf{z}_i\mathbf{z}_j \mid \mathbf{c}] - (\mathbf{s}\mathbf{c})_i(\mathbf{s}\mathbf{c})_j.$$

3 This is technically not true, but the experimental behaviour suggest that this assumption has no impact.

Rejection sampling makes an output $\mathbf{z}$ a centered discrete Gaussian of standard deviation σ. As $\mathbf{z}$'s coordinates are mutually independent, we get $\mathbb{V}_{ij} = (\mathbf{sc})_i(\mathbf{sc})_j$ if $i \neq j$ and $\mathbb{V}_{ii} = \sigma^2 - (\mathbf{sc})_i(\mathbf{sc})_j$. Recall that the coordinates of $\mathbf{sc}$ have magnitude at most κ, which is already a very pessimistic estimation. Hence $\mathbb{V}[(-1)^b\mathbf{z}]$ is essentially a scalar matrix $\sigma^2\mathbf{I}_n$. As $\mathbf{c}^\star$ has exactly κ non-zero entries, we get the desired result on $\mathbb{V}[\mathbf{w}]$.