IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 4, 6 pages.
https://doi.org/10.62056/a3c39qgxq

A Key-Recovery Attack on a Leaky SeaSign Variant

Shai Levin

University of Auckland, New Zealand

Abstract. We present a key-recovery attack on a variant of the SeaSign signature scheme presented by [Kim24], which attempts to avoid rejection sampling by presampling vectors $\mathbf{f}$ such that the $\mathbf{f} - \mathbf{e}$ is contained in an acceptable bound, where $\mathbf{e}$ is the secret key. We show that this choice leads to a bias of these vectors such that, in a small number of signatures, the secret key can either be completely recovered or its keyspace substantially reduced. In particular, given 20 signatures, with parameter set II of their paper, the attack reduces the private key to 128 possibilities.

Keywords: Isogeny-based cryptography · Cryptanalysis · CSIDH · Signature Schemes

1 Introduction

SeaSign is an isogeny group-action based signature scheme, first proposed by De Feo and Galbraith [DG19], and later refined by [DPV19]. The scheme is derived from a sigma-protocol for a proof of knowledge of a one-way function obtained by isogeny group-actions. The security of SeaSign relies on rejection sampling, in a Fiat-Shamir-with-aborts [Lyu09] setting, where the prover may restart the protocol in order to prevent leaking information about the secret. The core idea is to ensure that the responses sent by the signer, which are either ephemeral values or differences between ephemeral values and the secret key, remain within specific bounds. If a response falls outside these bounds, the signer aborts the protocol and restarts, thereby preventing any unintended leakage of information about the secret key.

However, the work by Kim [Kim24], introduces a variant of the SeaSign scheme that attempts to bypass the need for rejection sampling, eliminating the potential for unnecessary computations caused by protocol aborts and restarts. The proposed variant claims to achieve this by pre-sampling commitment vectors such that responses will be distributed uniformly for either challenge bit, independent of the secret key. However, we show that this approach inadvertently introduces a bias in the distribution of the responses. The signer's attempt to avoid rejection sampling by pre-sampling commitment vectors leads to a situation where certain responses become impossible, depending on entries of the secret key.

Notation

Given an indexed set of vectors, we use $\mathbf{v}_i^{(j)}$ to refer to the i-th entry of the j-th vector. We refer to the set $\{1 \dots n\}$ as $[n]$, and the set of integers between a and b (inclusive) as $[a, b]$.

E-mail: shai.levin@auckland.ac.nz (Shai Levin)

Received: 2024-10-02 Accepted: 2024-12-03

2 The SeaSign Variant in [Kim24]

Since the cryptanalysis of our work does not depend on the technical nature of isogeny-based group actions, we will not delve into the technical details of the SeaSign scheme. Instead, we will briefly describe the relevant features of SeaSign signature generation, and include a description of the variant proposed by Kim [Kim24].

The SeaSign scheme in [DG19]. SeaSign is based on an identification protocol where the secret key corresponds to a secret vector $\mathbf{e} \in [-B, B]^n$ which is used as input to a one-way function:

$$f(\mathbf{e}) = \mathfrak{i}_1^{\mathbf{e}_1} \dots \mathfrak{i}_n^{\mathbf{e}_n} \star E = E'$$

where E, E' are elliptic curves, and $\mathfrak{i}_i$ are elements of the ideal class group of $\mathrm{End}(E)$, which defines a group action on the set of supersingular curves over a finite field $\mathbb{F}_p$. The public key corresponds to E', and the signature is a proof of knowledge that the signer knows the secret key $\mathbf{e}$, with the message tied into the randomness of the challenge computation.

As part of the original protocol, the signer samples, and commits to, random vectors $\mathbf{f}^{(j)} \leftarrow\!\!\$ \; [-(\delta+1)B, (\delta+1)B]^n$ for $j \in \{1, \dots, t\}$. After receiving t single bit challenges, for each challenge c_j, the signer either sends $\mathbf{f}^{(j)}$ if $c_j = 0$ or $\mathbf{f}^{(j)} - \mathbf{e}$ if $c_j = 1$. However, for $c_j = 1$, the signer leaks some information about the vector $\mathbf{e}$, since the distribution of $\mathbf{f}^{(j)} - \mathbf{e}$ is not uniform in $[-(\delta+1)B, (\delta+1)B]^n$. To avoid this, rejection sampling is used. After computing the challenge, if a response vector, which is either $\mathbf{f}^{(j)}$ or $\mathbf{f}^{(j)} - \mathbf{e}$, is not in the bound $[-\delta B, \delta B]^n$, the signer aborts the protocol and restarts. This is repeated until the signer sends a valid response.

Modifications to Signature Generation in [DPV19]. The approach is refined in the follow up work [DPV19] where aborts are avoided in the case that $c_j = 0$, since this only reveals the ephemeral values $\mathbf{f}^{(j)}$, and leak nothing about the secret. Furthermore, given t iterations of the sigma protocol, their protocol tolerates up to u aborts (for $u < t$) before the protocol must be re-executed, which substantially improves signature generation efficiency.

Modifications to Signature Generation in [Kim24]. The approach of [Kim24] differs from these two prior works by attempting to avoid rejection sampling completely, by presampling commitment vectors $\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(t)}$ such that $\mathbf{f}^{(j)} - \mathbf{e} \in [-\delta B, \delta B]^n$. Since the signer would not need to abort and restart the protocol, this would prevent unnecessary isogeny computations, speeding up signing time. However, the key difference is that the signer cannot know what the challenge bits will be in advance, and cannot prevent bias in the distributions of the responses.

Algorithm 1 Signature Generation in [Kim24]

```
Input: message $m$, $pk = (E, E_A)$, secret key $\mathbf{e} \in [-B, B]^n$
Output: $\sigma = (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(t)}, c_1, \ldots, c_t)$    ▷ $\mathbf{z}^{(j)} \in [-(\delta+1)B, (\delta+1)B]^n, c_j \in \{0,1\}$
1: cnt ← 1
2: while cnt ≤ $t$ do
3:     $\mathbf{f}^{(\mathrm{cnt})} \leftarrow\!\!\$ \; [-(\delta+1)B, (\delta+1)B]^n$
4:     $b \leftarrow \mathbf{f}^{(\mathrm{cnt})} - \mathbf{e}$
5:     if $b \in [-\delta B, \delta B]^n$ then    ▷ Resample if out of acceptable bound
6:         $\mathbf{z}^{(\mathrm{cnt})} \leftarrow \mathbf{f}^{(\mathrm{cnt})}$
7:         $E_{\mathrm{cnt}} \leftarrow \mathfrak{l}_1^{\mathbf{f}_1^{(\mathrm{cnt})}} \ldots \mathfrak{l}_n^{\mathbf{f}_n^{(\mathrm{cnt})}} \star E$
8:         cnt ← cnt + 1
9:     end if
10: end while
11: $c_1, \ldots, c_t \leftarrow H(j(E_1), \ldots, j(E_t), m)$    ▷ Compute the challenge bits[1]
12: for $j$ from 1 to $t$ do
13:     if $c_j = 0$ then
14:         $\mathbf{z}^{(j)} \leftarrow \mathbf{z}^{(j)}$    ▷ The leaky case, if $c_j = 0$
15:     else
16:         $\mathbf{z}^{(j)} \leftarrow \mathbf{z}^{(j)} - \mathbf{e}$
17:     end if
18: end for
```

3 The Attack

We now state an attack on signatures generated by Algorithm 1. Suppose that we are given samples

$$\mathbf{f} \leftarrow\!\!\$ \; [-(\delta+1)B, (\delta+1)B]^n, \text{ such that } \mathbf{f} - \mathbf{e} \in [-\delta B, \delta B]^n$$

for a fixed, uniform secret $\mathbf{e} \in [-B, B]^n$. The first observation is that the i-th entries of sampled vectors are uniformly distributed in the set, i.e.

$$\mathbf{f}_i \leftarrow\!\!\$ \; [-\delta B + \mathbf{e}_i, \delta B + \mathbf{e}_i],$$

which is clearly a distribution dependent on the secret $\mathbf{e}$. In order to exploit this key dependence, our attack amounts to guessing the unknown upper and lower bounds for the distribution above. Suppose you are given m samples $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(m)}$. We define

$$a_i = \min_{j \in [m]} \left(\mathbf{f}_i^{(j)} + \delta B, B \right) \qquad b_i = \max_{j \in [m]} \left(\mathbf{f}_i^{(j)} - \delta B, -B \right) \tag{1}$$

Theorem 1. *Given m vectors $\{\mathbf{f}^{(j)}\}_{j \in [m]}$ such that, for all $j \in [m]$ and some fixed $\mathbf{e} \in [-B, B]^n$, it holds that $\mathbf{f}^{(j)} \in [-(\delta+1)B, (\delta+1)B]^n$ and $\mathbf{f}^{(j)} - \mathbf{e} \in [-\delta B, \delta B]^n$. Then for $i \in [n]$ and a_i's and b_i's computed as per Equation (1), we have that $\mathbf{e}_i \in [b_i, a_i]$. In particular, if $a_i = b_i$, then $\mathbf{e}_i = a_i = b_i$.*

Proof. Suppose that $\mathbf{e}_i \notin [b_i, a_i]$. Since $\mathbf{e}_i \in [-B, B]$, we consider either the case that:

$\mathbf{e}_i < b_i$: in which case $b_i \neq -B$ (since $\mathbf{e}_i \geq -B$), and there exists some maximal $\mathbf{f}_i^{(j)}$ such that $\mathbf{f}_i^{(j)} = b_i + \delta B$. Then $\mathbf{f}_i^{(j)} - \mathbf{e}_i = b_i + \delta B - \mathbf{e}_i > \delta B$, which contradicts the assumption that $\mathbf{f}^{(j)} - \mathbf{e} \in [-\delta B, \delta B]$.

[1] $j(E)$ is a function which returns the j-invariant of an elliptic curve E (given its curve coefficients over a finite field).

$\mathbf{e}_i > a_i$: in which case $a_i \neq B$ (since $\mathbf{e}_i \leq B$), and there exists some minimal $\mathbf{f}_i^{(j)}$ such that $\mathbf{f}_i^{(j)} = a_i - \delta B$. Then $\mathbf{f}_i^{(j)} - \mathbf{e}_i = a_i - \delta B - \mathbf{e}_i < -\delta B$, which contradicts the assumption that $\mathbf{f}^{(j)} - \mathbf{e} \in [-\delta B, \delta B]$. □

Hence the attack is as follows. On input s signatures $\sigma_1, \ldots, \sigma_s$:

1. From each signature, collect the set of vectors $\{z^{(j)}\}$ for which $c_j = 0$. Set m to be the size of such set.
2. For each $i \in [n]$, compute a_i and b_i as per Equation (1). Output the set of guesses for $\mathbf{e}$ as $\bigoplus_{i=1}^{n} [b_i, a_i]$.

Let $P_{m,i}$ be probability of m biased vectors leaking the i-th entry of the secret key, which satisfies the bound:

$$P_{m,i} \geq 1 - 2\left(1 - \frac{1}{2(\delta+1)B+1}\right)^m + \left(1 - \frac{2}{2(\delta+1)B+1}\right)^m$$

Hence the probability P_m of m biased vectors leaking the entire secret key satisfies $P_m = P_{m,i}^n$. The parameters in this setting are not well suited for approximating the binomial terms via low-order taylor approximations. For reference, however, with parameter set II; $P_{6173} \approx 0.5$. Hence, the attack is expected to recover the secret key in a small number of signatures. We provide practical results of the attack in the next section, which account for the keyspace reduction of the signing key.

On guessing the correct value of secret scalars By Theorem 1, given m samples for each entry $i \in [n]$, we determine some interval $[b_i, a_i]$ which contains $\mathbf{e}_i$. Fix a row i. While it might seem intuitive to prioritize values near the mean value of the interval $[b_i, a_i]$, we show that given samples $\mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)}$, the probability that they result from the distribution $[-\delta B + \mathbf{e}_i, \delta B + \mathbf{e}_i]$ is uniform for the choice of $\mathbf{e}_i$ over $[b_i, a_i]$. By construction of Equation (1), we have that $\mathbf{f}_i^{(j)} \in [-\delta B + a_i, \delta B + b_i]$ for all $j \in [m]$. This interval is strictly contained in $[-\delta B + \mathbf{e}_i, \delta B + \mathbf{e}_i]$ if and only if $\mathbf{e}_i \in [b_i, a_i]$. Let $\mathcal{D}_{\mathbf{e}_i}$ be the uniform distribution on $[-\delta B + \mathbf{e}_i, \delta B + \mathbf{e}_i]$. Hence, the event of receiving the observed samples $\mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)}$ arise from the distribution $\mathcal{D}_{\mathbf{e}_i}$ precisely once when $\mathbf{e}_i \in [b_i, a_i]$ (and cannot occur otherwise). Now, observe that for all $\mathbf{e}_i \in [b_i, a_i]$, it holds that:

$$\begin{aligned}\Pr[\mathbf{e}_i \mid \mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)}] &= \frac{\#\text{Events where } \mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)} \text{ arises from } \mathcal{D}_{\mathbf{e}_i}}{\#\text{Events where } \mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)} \text{arises from } \mathcal{D}_x \text{ for some } x \in [b_i, a_i]} \\ &= \frac{\#\text{Events where } \mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)} \text{ arises from } \mathcal{D}_{\mathbf{e}_i}}{(\#\text{Events where } \mathbf{f}_i^{(1)}, \ldots, \mathbf{f}_i^{(m)} \text{arises from } \mathcal{D}_x) \cdot (\#x \in [b_i, a_i])} \\ &= \frac{1}{1 \cdot (a_i - b_i + 1)}.\end{aligned}$$

This implies there is no better way to guess the correct value of $\mathbf{e}_i$ than to guess uniformly over the interval $[b_i, a_i]$, indicating the attack is in a sense, optimal.

On avoiding rejection sampling At a high level, the reason why rejection sampling after the challenge computation cannot be avoided, is that the distribution of responses to a fixed challenge bit must be independent of the secret in both cases. By forcing a bound on either $\mathbf{f}$ or $\mathbf{f} - \mathbf{e}$ prior to knowing which challenge bit is chosen, the signer introduces a bias in the distribution of the responses. In our case, $\mathbf{f} - \mathbf{e}$ was rejected if out of bounds, which induced key dependence to the distributions of responses $\mathbf{f}$ for $c = 0$, but indeed the same issue would occur for responses $\mathbf{f} - \mathbf{e}$ to $c = 1$ if instead the resampling was performed when $\mathbf{f}$ did not satisfy some bound. The signer cannot know the challenge bits in advance, and hence can only perform rejection of leaky responses after challenge computation.

	n	B	δ	t
Parameter Set I [DG19]	74	5	9472	128
Parameter Set II [DPV19]	74	5	114	337

Figure 1: Parameter sets used in [Kim24].

4 Implementation and Benchmarks

We implement the key-recovery attack using a sage script available at `https://github.com/levanin/leakysea-public`. On each iteration of the experiment, a random key is sampled and a fixed number of biased samples are generated. The protocol of [Kim24] only leaks a biased vector when a challenge bit is 0, which occurs with probability $\frac{1}{2}$. So we will assume that given s signatures with challenge length t, we may obtain $\lfloor \frac{st}{2} \rfloor$ biased vectors.

The attack is efficient, and all of our benchmarking was comfortably performed on a laptop over a lunch break. We provide the results of our attack given a varying number of signatures on the parameter sets provided by [Kim24] in Figures 2 and 3, obtained from the prior works [DPV19, DG19]. Once the keyspace has been reduced to a size 2^b, a meet-in-the-middle search strategy can be used to recover the secret key in time $O(2^{b/2})$, using techniques described in [DG19].

We note that the parameter set I requires a larger number of signatures to effectively perform the attack. This parameter set is designed to handle the high failure probability of the original SeaSign protocol, so ephemeral vectors must be sampled from a much larger space. Hence, there is a lower chance of receiving "good" vectors which leak information about the secret key. We remark that it would have be unreasonable to use these parameters over parameter set II in the first place, since they do not yield any performance benefits over existing work. In particular, the performance of the prior work [DPV19] using parameter set II is roughly $10\times$ faster ($2,195$ s) than the performance of [Kim24] running on parameter set I ($27,685.92$ s), with claimed equivalent security levels.

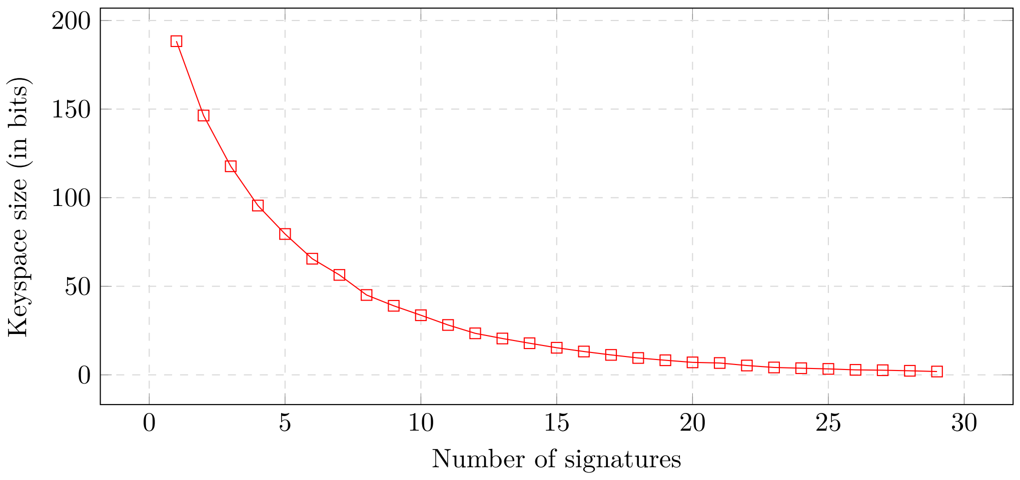

Figure 2: Results of our attack on [Kim24, Parameter Set II]. Results are the mean over 100 random instances. The keyspace refers to the bit-length of the size of the set of possible secret keys (i.e., if the keyspace is n bits, then the number of possible secret keys is 2^n).

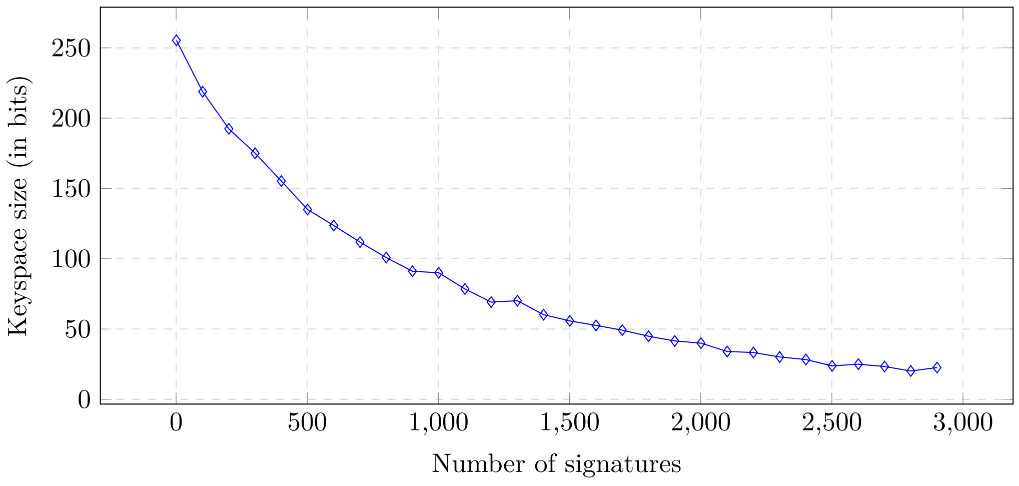

Figure 3: Results of our attack on [Kim24, Parameter Set I]. Results are the mean over 20 random instances.

References

[DG19] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Cham, May 2019. doi:10.1007/978-3-030-17659-4_26.

[DPV19] Thomas Decru, Lorenz Panny, and Frederik Vercauteren. Faster SeaSign signatures through improved rejection sampling. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 271–285. Springer, Cham, 2019. doi:10.1007/978-3-030-25510-7_15.

[Kim24] Suhri Kim. Optimized SeaSign for Enhanced Efficiency. *IEEE Access*, pages 1–1, 2024. URL: https://ieeexplore.ieee.org/document/10416853/, doi:10.1109/ACCESS.2024.3360297.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Berlin, Heidelberg, December 2009. doi:10.1007/978-3-642-10366-7_35.