From: pqc-forum@list.nist.gov on behalf of Mehdi Tibouchi <mehdi.tibouchi@normalesup.org>
Sent: Monday, July 17, 2023 12:32 PM
To: pqc-comments
Cc: pqc-forum
Subject: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign

Dear EagleSign submitters, dear all,

The EagleSign signature is a Fiat-Shamir construction without aborts that uses small noise and a polynomially-bounded modulus; as such, it is not plausibly secure.

Leakage of the secret key in signatures occurs for example in signature element z, which is of the form:

z = G·(y_1 + c)

where G is a short matrix part of the secret key (and recovering it is effectively sufficient for full key recovery), c is known and small, and the y_1 is small with known, independent distribution. No modular reduction occurs in this equation.

As a result, the distribution of z depends on G in a way that easily reveals it using statistical techniques (for example, collect many signatures for which the first coefficient of c is 1, which is satisfied by a significant fraction of all signatures, and observe that the conditional expectation depends linearly on G). The same attack can be mounted on the other signature element w.

Accordingly, the security proof is incorrect. For example, the element z of the signature is simulated as a uniformly random among the elements of R of infinity norm up to the maximum possible according to the equation above, but clearly, the real distribution is not uniform at all.

Best regards,

--
M. Tibouchi
<mehdi.tibouchi@normalesup.org>

****
NTT Social Informatics Laboratories
Abe Research Laboratory
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan.
****

From: pqc-forum@list.nist.gov on behalf of Mehdi Tibouchi <mehdi.tibouchi@normalesup.org>
Sent: Wednesday, July 19, 2023 2:23 AM
To: pqc-comments
Cc: pqc-forum
Subject: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign

Dear EagleSign submitters, dear all,

For the sake of completeness, you can find sample code that implements the attack below in the following GitHub repository:

https://github.com/mti/attack_eaglesign

As mentioned in the README, this is a fairly naive implementation of the idea, but it already correctly recovers around 1020 coefficients of G out of 1024 for parameter set EagleSign-3 with 100,000 signature samples on arbitrary messages.

Best regards,

--
M. Tibouchi
<mehdi.tibouchi@normalesup.org>
****
NTT Social Informatics Laboratories
Abe Research Laboratory
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan.
****

On Mon, Jul 17, 2023 at 06:32:03PM +0200, Mehdi Tibouchi wrote:

> Dear EagleSign submitters, dear all,
>
> The EagleSign signature is a Fiat-Shamir construction without aborts that
> uses small noise and a polynomially-bounded modulus; as such, it is not
> plausibly secure.
>
> Leakage of the secret key in signatures occurs for example in
> signature element z, which is of the form:
>
> $z = G\cdot(y_1 + c)$
>
> where G is a short matrix part of the secret key (and recovering it is
> effectively sufficient for full key recovery), c is known and small, and
> the y_1 is small with known, independent distribution. No modular

From: Ludo Pulles <lnp@cwi.nl>
Sent: Wednesday, July 19, 2023 5:20 AM
To: Mehdi Tibouchi; pqc-comments
Cc: pqc-forum
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign

Hi Mehdi,

Great work finding this. I think the outlined attack is already quite disastrous. Just to remove any doubt whether recovering all but a couple of the coefficients really breaks the scheme:

https://github.com/ludopulles/EagleHasFlown

Here is an implementation that throws away less of the signatures (re: your 2nd remark in the README): you can use the same statistical argument for all the other coefficients of c as well, greatly reducing the number of (random) signatures required for the attack to work.
The matrix G is recovered completely with 300 and 500 signatures for EagleSign3 and EagleSign5 respectively, and the matrix D is fully recovered with 2000 signatures for both EagleSign-{3,5}.

Best regards,
Ludo

> **Van:** Mehdi <mehdi.tibouchi@normalesup.org>
> **naar:** pqc-comments <pqc-comments@nist.gov>
> **CC:** pqc-forum <pqc-forum@list.nist.gov>
> **datum:** woensdag 19 juli 2023 8:23 CEST
> **Onderwerp:** [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign
>
> Dear EagleSign submitters, dear all,
>
> For the sake of completeness, you can find sample code that implements
> the attack below in the following GitHub repository:
>
> https://github.com/mti/attack_eaglesign
>
> As mentioned in the README, this is a fairly naive implementation of the
> idea, but it already correctly recovers around 1020 coefficients of G out
> of 1024 for parameter set EagleSign-3 with 100,000 signature samples on
> arbitrary messages.
>
> Best regards,
>
> --
> M. Tibouchi
> <mehdi.tibouchi@normalesup.org>

From: djiby sow <djiby.sow.pr@gmail.com>
Sent: Wednesday, July 19, 2023 1:55 PM
To: Mehdi Tibouchi
Cc: pqc-comments; pqc-forum
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign

Hi all
Great work Tibouchi for this attack on EagleSign and for your Quick implementation of this attack . Impressive.
Since EagleSign involve other variants, we are testing the zero knowledge property for these variants
Best regards .

Le mer. 19 juil. 2023 à 06:23, Mehdi Tibouchi <mehdi.tibouchi@normalesup.org> a écrit :

> Dear EagleSign submitters, dear all,
>
> For the sake of completeness, you can find sample code that implements the attack below in the following GitHub repository:
>
> https://github.com/mti/attack_eaglesign
>
> As mentioned in the README, this is a fairly naive implementation of the idea, but it already correctly recovers around 1020 coefficients of G out of 1024 for parameter set EagleSign-3 with 100,000 signature samples on arbitrary messages.
>
> Best regards,
>
> --
> M. Tibouchi
> <mehdi.tibouchi@normalesup.org>
> http://www.normalesup.org/~tibouchi/
> ***
> NTT Social Informatics Laboratories
> Abe Research Laboratory
> 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan.
> ***
>
> On Mon, Jul 17, 2023 at 06:32:03PM +0200, Mehdi Tibouchi wrote:
> > Dear EagleSign submitters, dear all,
> >
> > The EagleSign signature is a Fiat-Shamir construction without aborts that
> > uses small noise and a polynomially-bounded modulus; as such, it is not
> > plausibly secure.
> >
> > Leakage of the secret key in signatures occurs for example in
> > signature element z, which is of the form:
> >
> >   z = G·(y_1 + c)
> >

From: djiby sow <djiby.sow.pr@gmail.com>
Sent: Wednesday, July 19, 2023 2:04 PM
To: Ludo Pulles
Cc: Mehdi Tibouchi; pqc-comments; pqc-forum
Subject: Re: [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign

Hi all
Great work Pulles for your work by improving the implementation of the attack of Tibouchi on EagleSign.
As said before we are working other variants that hold the zero knowledge property
Best regards

Le mer. 19 juil. 2023 à 09:19, 'Ludo Pulles' via pqc-forum <pqc-forum@list.nist.gov> a écrit :

> Hi Mehdi,
>
> Great work finding this. I think the outlined attack is already quite disastrous. Just to remove any doubt whether recovering all but a couple of the coefficients really breaks the scheme:
>
> https://github.com/ludopulles/EagleHasFlown
>
> Here is an implementation that throws away less of the signatures (re: your 2nd remark in the README): you can use the same statistical argument for all the other coefficients of c as well, greatly reducing the number of (random) signatures required for the attack to work.
> The matrix G is recovered completely with 300 and 500 signatures for EagleSign3 and EagleSign5 respectively, and the matrix D is fully recovered with 2000 signatures for both EagleSign-{3,5}.
>
> Best regards,
> Ludo
>
>> **Van:** Mehdi <mehdi.tibouchi@normalesup.org>
>> **naar:** pqc-comments <pqc-comments@nist.gov>
>> **CC:** pqc-forum <pqc-forum@list.nist.gov>
>> **datum:** woensdag 19 juli 2023 8:23 CEST
>> **Onderwerp:** [pqc-forum] Re: Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign
>>
>> Dear EagleSign submitters, dear all,
>>
>> For the sake of completeness, you can find sample code that implements
>> the attack below in the following GitHub repository:
>>
>> https://github.com/mti/attack_eaglesign
>>
>> As mentioned in the README, this is a fairly naive implementation of the
>> idea, but it already correctly recovers around 1020 coefficients of G out
>> of 1024 for parameter set EagleSign-3 with 100,000 signature samples on
>> arbitrary messages.

From: pqc-forum@list.nist.gov on behalf of Abiodoun HOUNKPEVI <abiodounkpevi@gmail.com>
Sent: Sunday, December 10, 2023 9:02 AM
To: pqc-forum
Subject: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign-V2, A New improvement of EagleSign with Zero Knowledge property

Dear All,

In response to the recent attack proposed by Tibouchi on the NIST forum regarding the security of EagleSign signatures here, we have undertaken significant enhancements to fortify the protocol against the proposed attack (A similaire attack is described in Agrawal, S., Stehlé, D., & Yadav, A. Round-optimal lattice-based threshold signatures, revisited. *Cryptology ePrint Archive 2022* \textit{https://eprint.iacr.org/2022/634}) .

The updated package is available through the link here.

The vulnerabilities identified, particularly in the leakage of the secret key within signature elements, have been addressed through the implementation of two key techniques:

1. **Change of the particular Public Key**:

To bolster the security of EagleSign, we have revised the public key generation process. Our general public key was $E = (AF^{-1} + D)\cdot G^{-1}$ (zere G is a matrix or a polynomial) but for efficiency we have chosen $E = (A + D)\cdot G^{-1}$ as the public key in our particular instantiation in EagleSign submitted to NIST. Now, the modification involves transitioning from the previous formulation $E = (A + D)\cdot G^{-1}$ to a more resilient expression $E = (A\cdot F + D)\cdot g^{-1}$ where A represents the public matrix generated from a seed, g represents a sparse and small invertible polynomial, D represents a sparse and small polynomial matrix, F is an invertible, small, and uniformly generated polynomial matrix, This adjustment aims to be able to change the mathematical formulas of the signature and introduce rejection sampling in order mitigate the specific vulnerability associated with the short matrix part G of the secret key.

2. **Change in the Signature:**
2.1 Change of the mathematical formulas:
The signature generation process has undergone a fundamental change to counteract potential attacks. We have moved from the susceptible equation $z = G\cdot(y_1 + c)$ to a more secure formulation: $z = g\cdot(y + F\cdot c)$. Here, y is a randomly generated vector in a very large interval, and c is a challenge vector consisting of small values.

2.2 **Introduction of Rejection Sampling Method:**
To further control the size of the signature and mitigate potential threats, we have integrated the rejection sampling method proposed by Lyubachevski. This method allows for the selective rejection of certain signatures, enhancing overall security.

Conclusion
These alterations disrupt the linear relationship between key components and enhance resistance to statistical techniques since z does not depend on g anymore, but instead on y which is randomly chosen. Since then, our signature does not leak private keys, hence we have the zero knowledge property as desired.
These strategic modifications collectively address the concerns raised by Tibouchi and significantly strengthen the security posture of EagleSign-V2.

The result of these is that the modulus q increases, hence the size of the public key is more big but the size of the signature is similar to those of Dilithium..

To reduce the size of the public key we are working now on a new variant . The old variant is being implemented over $R_q=\dfrac{\mathbb{Z}_q (X)}{(X^n+1)}$ with $n=1024, 2048$ in order to make NTT easy to use. To reduce the size of the public key one can choose the ring $R_q=\dfrac{\mathbb{Z}_q (X)}{(X^n+X^{n/2}-1)}$ with $n=768, 1536$ (where we can use also NTT), this ring is studied in "Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D. (2023). \emph{A Thorough Treatment of Highly-Efficient NTRU Instantiations}. In: Boldyreva, A., Kolesnikov, V. (eds) Public-Key Cryptography – PKC 2023. PKC 2023. Lecture Notes in Computer Science, vol 13940. Springer, Cham. https://doi.org/10.1007/978-3-031-31368-4_ 3"

Best regards,

On behalf of EagleSign Team,
Clement HOUNKPEVI
--

From: pqc-forum@list.nist.gov on behalf of Mehdi Tibouchi <mehdi.tibouchi@normalesup.org>
Sent: Sunday, December 10, 2023 12:28 PM
To: Abiodoun HOUNKPEVI
Cc: pqc-forum
Subject: Re: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign-V2, A New improvement of EagleSign with Zero Knowledge property

Dear submitters, dear all,

Unless I am missing something, the updated scheme can be broken as follows.

First, note that there is again no modular reduction in the computation of u = y + Fc and z = gu, since all the elements involved are small. So these are equalities over the ring.

As a result, the ideal gR can easily be recovered as the sum of the ideals zR for the values z in a handful of signatures. Then, g can be recovered (up to a root of unity) in various ways. I'm sure one can think of better approaches, but meet-in-the-middle attacks are sufficient to break the security claims in view of the the very low entropy: for "NIST-2" parameters, the entropy is only ~ 2^56 after quotienting by roots of unity, so the complexity of MitM is only ~ 2^28 for the most naive approach (similarly for "NIST-5", the numbers are ~ 2^123 and ~ 2^62).

Once g is recovered, a statistical attack on u recovers F (note that the rejection sampling does not work as stated due to the fact that, for w in S_{t_g B}, it is not the case that g^{-1}w is necessarily in S_{B}, and hence the argument of section 3.2 is flawed), and thus the entire key.

The scheme can probably be fixed by replacing the signature generation again by a proper variant of Lyubashevsky'12, but at that point it would presumably just be a less efficient version of ML-DSA, so I fail to get the selling point.

Best regards,

--
M. Tibouchi
<mehdi.tibouchi@normalesup.org>
****
NTT Social Informatics Laboratories
Abe Research Laboratory
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan.
****

On Sun, Dec 10, 2023 at 03:01:40PM +0100, Abiodoun HOUNKPEVI wrote:

> Dear All,
>
> In response to the recent attack proposed by Tibouchi on the NIST forum

From: pqc-forum@list.nist.gov on behalf of Mehdi Tibouchi <mehdi.tibouchi@normalesup.org>
Sent: Wednesday, December 13, 2023 2:57 AM
To: Abiodoun HOUNKPEVI
Cc: pqc-forum
Subject: Re: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign-V2, A New improvement of EagleSign with Zero Knowledge property

Dear submitters, dear all,

Regarding the attack below, let me be a bit more precise.

A simple observation is that the rejection condition:

  |z|_\infty >= t_g (gamma_1 - beta)

is essentially never triggered, as can be immediately verified on the provided implementation. This is because the coefficients of the "randomness term" in z = g·u, namely g·y, are distributed like sums of t_g samples from the uniform distribution in [-gamma_1, gamma_1], and therefore concentrate a lot more around 0 than the uniform distribution in [-t_g·gamma, t_g·gamma].

As a result, the newly introduced rejection sampling basically doesn't affect the distribution. (The other two rejection conditions can be triggered, but aren't really relevant to the behavior of z, since E_s looks uniform mod q). This means that the statistically attack on the previous version of EagleSign applies without modification to the new scheme, but instead recovers gF, and after than DF = E_s gF - A.

This is sufficient to show that the claimed HVZK property doesn't hold.
For a full key recovery, one can do as suggested below, and carry out a meet-in-the-middle attack on g, which combined with the previous elements reveals the entire key.

Best regards,

--
Mehdi.

On Sun, Dec 10, 2023 at 06:27:35PM +0100, Mehdi Tibouchi wrote:
> Dear submitters, dear all,
>
> Unless I am missing something, the updated scheme can be broken as
> follows.
>
> First, note that there is again no modular reduction in the
> computation of u = y + Fc and z = gu, since all the elements involved
> are small. So these are equalities over the ring.
>
> As a result, the ideal gR can easily be recovered as the sum of the
> ideals zR for the values z in a handful of signatures. Then, g can be
> recovered (up to a root of unity) in various ways. I'm sure one can
> think of better approaches, but meet-in-the-middle attacks are

From: pqc-forum@list.nist.gov on behalf of djiby sow <djiby.sow.pr@gmail.com>
Sent: Thursday, December 21, 2023 2:22 PM
To: Mehdi Tibouchi
Cc: Abiodoun HOUNKPEVI; pqc-forum
Subject: Re: [pqc-forum] Round 1 (Additional Signatures) OFFICIAL COMMENT: EagleSign-V2, A New improvement of EagleSign with Zero Knowledge property

Dear all
,Dear tibouchi, your security analysis of EagleSgn is wonderful. Your hard work allows us to improve prograssively our submission to NIST.

Dear all, we agree with Tibouchi that the formula used in the signature (not the public key) contains a flaw. Tibouchi remarks that the deal $\mathbf{G}R_q$ can be easily recovered as the sum of the ideals $\mathbf{Z}R_q$ for the values $\mathbf{Z}$ in a handful of signature" and therefore $\mathbf{G}$ can be recovered. This fact induce also a flaw in the section 3.2 for the zero knowledge propertry.
Currently we are working to modify slightly the formula of the signature to circumvent these weakness. The new variant will be available in few days.
Best regards
On behalf of EagleSign team
Pr djiby Sow

Le mer. 13 déc. 2023 à 07:57, Mehdi Tibouchi <mehdi.tibouchi@normalesup.org> a écrit :

> Dear submitters, dear all,
>
> Regarding the attack below, let me be a bit more precise.
>
> A simple observation is that the rejection condition:
>
> |z|_\infty >= t_g (gamma_1 - beta)
>
> is essentially never triggered, as can be immediately verified on the provided implementation. This is because the coefficients of the "randomness term" in z = g·u, namely g·y, are distributed like sums of t_g samples from the uniform distribution in [-gamma_1, gamma_1], and therefore concentrate a lot more around 0 than the uniform distribution in [-t_g·gamma, t_g·gamma].
>
> As a result, the newly introduced rejection sampling basically doesn't affect the distribution. (The other two rejection conditions can be triggered, but aren't really relevant to the behavior of z, since E_s looks uniform mod q). This means that the statistically attack on the previous version of EagleSign applies without modification to the new scheme, but instead recovers gF, and after than DF = E_s gF - A.
>
> This is sufficient to show that the claimed HVZK property doesn't hold. For a full key recovery, one can do as suggested below, and carry out a meet-in-the-middle attack on g, which combined with the previous elements reveals the entire key.
>
> Best regards,