Overfull: Too Large Aggregate Signatures Based on Lattices

Katharina Boudgoust[1] Adeline Roux-Langlois[2]

[1]Aarhus University, Denmark

[2]IRISA, CNRS, Univ Rennes, France

CFAIL, Santa Barbara, 13th August 2022

1

Too Large Aggregate Signatures Based on Lattices

3

CFAIL

2

Digital Signatures (Informal)

Digital Signatures (Informal)

Motivation:

- Digital analogue of handprint signature
- Even more secure
- Even more functionalities

today

Digital Signatures (Formal)

$\Pi_S = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$

message 🗎

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}$

🗎, ✎ $= \mathsf{Sig}(\mathsf{sk},$ 🗎$)$ ⟶

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk},$ 🗎, ✎$)$

Signature is **valid** if $1 \leftarrow \mathsf{Vf}$.

Properties

Correctness

Unforgeability

Applications

Authentication

Multiple Signatures

message $🖹_1$

$(\color{orange}{\mathsf{sk}_1}, \color{cyan}{\mathsf{vk}_1}) \leftarrow \mathsf{KGen}$

$$\xrightarrow{🖹_1,\, ✎_1 \,=\, \mathsf{Sig}(\color{orange}{\mathsf{sk}_1}, 🖹_1)}$$

$\{0,1\} \leftarrow \mathsf{Vf}(\color{cyan}{\mathsf{vk}_1}, 🖹_1, ✎_1)$

Multiple Signatures

message 🗎$_1$

$(\textcolor{orange}{\mathsf{sk}_1}, \textcolor{cyan}{\mathsf{vk}_1}) \leftarrow \mathsf{KGen}$

$$\xrightarrow{\text{🗎}_1,\ \text{✎}_1 = \mathsf{Sig}(\textcolor{orange}{\mathsf{sk}_1}, \text{🗎}_1)}$$

message 🗎$_2$

$(\textcolor{orange}{\mathsf{sk}_2}, \textcolor{cyan}{\mathsf{vk}_2}) \leftarrow \mathsf{KGen}$

$\{0,1\} \leftarrow \mathsf{Vf}(\textcolor{cyan}{\mathsf{vk}_1}, \text{🗎}_1, \text{✎}_1)$

$$\xrightarrow{\text{🗎}_2,\ \text{✎}_2 = \mathsf{Sig}(\textcolor{orange}{\mathsf{sk}_2}, \text{🗎}_2)}$$

$\{0,1\} \leftarrow \mathsf{Vf}(\textcolor{cyan}{\mathsf{vk}_2}, \text{🗎}_2, \text{✎}_2)$

Multiple Signatures

message 🗎_1

$(\textcolor{orange}{\mathsf{sk}_1}, \textcolor{cyan}{\mathsf{vk}_1}) \leftarrow \mathsf{KGen}$

$$\xrightarrow{\text{🗎}_1,\ \text{✎}_1 = \mathsf{Sig}(\mathsf{sk}_1, \text{🗎}_1)}$$

message 🗎_2

$(\mathsf{sk}_2, \mathsf{vk}_2) \leftarrow \mathsf{KGen}$

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_1, \text{🗎}_1, \text{✎}_1)$

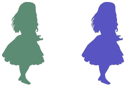
$$\xrightarrow{\text{🗎}_2,\ \text{✎}_2 = \mathsf{Sig}(\mathsf{sk}_2, \text{🗎}_2)}$$

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_2, \text{🗎}_2, \text{✎}_2)$

Q: Can we combine both ✎_1 and ✎_2 into a single signature?

And more generally for $N \gg 2$ many signatures?

Aggregate Signatures: AggSig and AggVf [BGLS03]



✎$_j$ $=$ Sig(sk_j, 🗎$_j$) for $j = 1, 2$

$\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$

✎ $\leftarrow$ AggSig(vk, 🗎$_1$, 🗎$_2$, ✎$_1$, ✎$_2$)

🗎$_1$,🗎$_2$,✎ $\longrightarrow$

$\{0, 1\} \leftarrow$ AggVf(vk, 🗎$_1$, 🗎$_2$, ✎)

Aggregate Signatures: AggSig and AggVf [BGLS03]

$\text{✎}_j = \mathsf{Sig}(\mathsf{sk}_j, \text{🗎}_j)$ for $j = 1, 2$

$\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$

$\text{✎} \leftarrow \mathsf{AggSig}(\mathsf{vk}, \text{🗎}_1, \text{🗎}_2, \text{✎}_1, \text{✎}_2)$

$$\xrightarrow{\text{🗎}_1, \text{🗎}_2, \text{✎}}$$

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, \text{🗎}_1, \text{🗎}_2, \text{✎})$

Properties

Correctness

Unforgeability

Compactness

Public aggregation

Applications

Consensus Protocols

Certificate Chains

Aggregate Signatures: AggSig and AggVf [BGLS03]

✎$_j$ = Sig(sk$_j$, 📄$_j$) for $j = 1, 2$

vk = (vk$_1$, vk$_2$)

✎ ← AggSig(vk, 📄$_1$, 📄$_2$, ✎$_1$, ✎$_2$)

$\xrightarrow{\text{📄}_1, \text{📄}_2, \text{✎}}$

only public input

$\{0, 1\}$ ← AggVf(vk, 📄$_1$, 📄$_2$, ✎)

Properties

Correctness

Unforgeability

Compactness

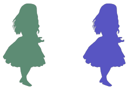
Public aggregation

Applications

Consensus Protocols

Certificate Chains

Aggregate Signatures: AggSig and AggVf [BGLS03]



✏$_j$ $=$ Sig(sk_j, 🗎$_j$) for $j = 1, 2$

$\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$

✏ $\leftarrow$ AggSig(vk, 🗎$_1$, 🗎$_2$, ✏$_1$, ✏$_2$)

🗎$_1$, 🗎$_2$, ✏ $\longrightarrow$

$\{0, 1\} \leftarrow$ AggVf(vk, 🗎$_1$, 🗎$_2$, ✏)

Properties

Correctness

Unforgeability

Compactness

Public aggregation

hard to obtain!

Research Question (from Oct 2020):

Can we construct an aggregate signature scheme

based on **Euclidean lattices?**

Research Question (from Oct 2020):

Can we construct an aggregate signature scheme

based on **Euclidean lattices?**

Yes, but...

- we need to be careful with the proofs (**pre-failed twice**) and
- our aggregate signature is larger than the concatenation of independent signatures (**final fail**)

1

Too Large Aggregate Signatures Based on Lattices

3

CFAIL

2

Fiat-Shamir with Aborts Signature [Lyu12]

Let $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = R/qR$ and $A' \leftarrow U(R_q^{k \times \ell})$ defining $A = [A' | I_k]$ and $H \colon \{0,1\}^* \to C \subseteq R$ be a random oracle

Fiat-Shamir with Aborts Signature [Lyu12]

Let $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = R/qR$ and $A' \leftarrow U(R_q^{k\times\ell})$ defining $A = [A'|I_k]$ and $H\colon \{0,1\}^* \to C \subseteq R$ be a random oracle

n power of 2, q prime

k and ℓ small constants

Fiat-Shamir with Aborts Signature [Lyu12]

Let $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = R/qR$ and $A' \leftarrow U(R_q^{k \times \ell})$ defining $A = [A'|I_k]$ and $H \colon \{0,1\}^* \to C \subseteq R$ be a random oracle

KGen

$\mathsf{sk} = s \leftarrow R^{k+\ell}$ small

$\mathsf{vk} = t = As \mod q$

Sig

$y \leftarrow R^{k+\ell}$ small, $u = Ay \mod q$

$c = H(u, , t) \in C \subseteq R$ small

$z = s \cdot c + y$ (rejection sampling)

$$\xrightarrow{\ , \ = (u, z)}$$

Vf

if $Az \stackrel{?}{=} t \cdot H(u, , t) + u$

and z small, accept

Fiat-Shamir with Aborts Signature [Lyu12]

Let $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = R/qR$ and $A' \leftarrow U(R_q^{k\times\ell})$ defining $A = [A'|I_k]$ and $H\colon \{0,1\}^* \to C \subseteq R$ be a random oracle

 📄

KGen $\mathsf{sk} = s \leftarrow R^{k+\ell}$ small

$\mathsf{vk} = t = As \mod q$

Sig $y \leftarrow R^{k+\ell}$ small, $u = Ay \mod q$

$c = H(u, 📄, t) \in C \subseteq R$ small

$z = s \cdot c + y$ (rejection sampling)

$$\xrightarrow{📄,\ ✏ = (u,z)}$$

Correctness:

$$\begin{aligned} & Az \\ = & A(sc + y) \\ = & (As)c + Ay \\ = & t \cdot H(u, 📄, t) + u \end{aligned}$$

Vf

if $Az \stackrel{?}{=} t \cdot H(u, 📄, t) + u$

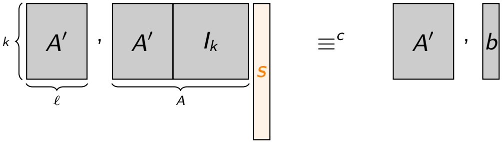
and z small, accept ✏

Unforgeability Based on Lattices

Theorem ([Lyu12])

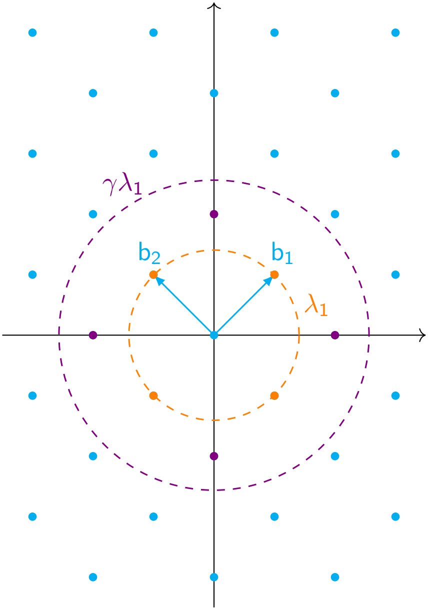
Assuming the hardness of the lattice problem Module LWE, the signature is secure against forgeries.

Module Learning With Errors (Module LWE): Distinguish



where $s \leftarrow R^{\ell+k}$ small and $(A', b) \leftarrow U(R_q^{k\times\ell} \times R_q^k)$.

- Presumably post-quantum secure
- Strong security guarantees
- Many cryptographic applications


$\gamma\lambda_1$
b_2
b_1
λ_1

News from July 2022

The same blueprint is used for the signature scheme **Dilithium** [DKL+18], which will be standardized by NIST!

Report 07/22

1

Too Large Aggregate Signatures Based on Lattices

3

CFAIL

2

Public Aggregation - First Attempt

 📄$_1$

KGen: $\mathsf{sk}_1 = s_1, \mathsf{vk}_1 = t_1 = As_1$

Sig:

$u_1 = Ay_1$

$c_1 = H(u_1, 📄_1, t_1)$

$z_1 = s_1 c_1 + y_1$ (rej. sampling)

✎$_1 = (u_1, z_1)$

 📄$_2$

$\mathsf{sk}_2 = s_2, \mathsf{vk}_2 = t_2 = As_2$

$u_2 = Ay_2$

$c_2 = H(u_2, 📄_2, t_2)$

$z_2 = s_2 c_2 + y_2$ (rej. sampling)

✎$_2 = (u_2, z_2)$

💡 Naive idea: ✎ $= (u, z) = (u_1 + u_2, z_1 + z_2)$ Vf $Az = t_1 c_1 + t_2 c_2 + u$

Public Aggregation - First Attempt

 📄$_1$

 📄$_2$

KGen

$\mathsf{sk}_1 = s_1, \mathsf{vk}_1 = t_1 = As_1$

$\mathsf{sk}_2 = s_2, \mathsf{vk}_2 = t_2 = As_2$

Sig

$u_1 = Ay_1$

$c_1 = H(u_1,$ 📄$_1, t_1)$

$z_1 = s_1 c_1 + y_1$ (rej. sampling)

✎$_1 = (u_1, z_1)$

$u_2 = Ay_2$

$c_2 = H(u_2,$ 📄$_2, t_2)$

$z_2 = s_2 c_2 + y_2$ (rej. sampling)

✎$_2 = (u_2, z_2)$

💡 Naive idea: ✎ $= (u, z) = (u_1 + u_2, z_1 + z_2)$ Vf $Az = t_1 c_1 + t_2 c_2 + u$

✖ Problem: How to compute c_1, c_2? Verifier doesn't know u_1, u_2

⚙ Half-aggregation: ✎ $= (u_1, u_2, z)$, $z = z_1 + z_2$ (as for Dlog analog)

Half-Aggregation - Fail!

Trick:

$$q \approx 2^{23}$$

Single signature: $✎ = (\underset{\in \{-q/2, \cdots, q/2\}^{nk}}{u}, z)$ Verification: $Az = t \cdot H(u, 🗎, t) + u$

Smaller signature: $✎ = (\underset{\in \{-1,0,1\}^n}{c}, z)$ Verification: $c = H(Az - tc, 🗎, t)$

This works only if the rabbit knows z.
Same trick not possible in the aggregate setting!

Half-Aggregation - Fail!

Trick:

$q \approx 2^{23}$

$u \in \{-q/2, \cdots, q/2\}^{nk}$

Single signature: ✎ $= (u, z)$ Verification: $Az = t \cdot H(u,$ 🗎 $, t) + u$

Smaller signature: ✎ $= (c, z)$ Verification: $c = H(Az - tc,$ 🗎 $, t)$

$c \in \{-1, 0, 1\}^n$

This works only if the rabbit knows z.
Same trick not possible in the aggregate setting!

Half-aggregation: ✎ $= (u_1, u_2, z_1 + z_2)$

Trivial: ✎ $= (c_1, c_2, z_1, z_2)$

Size: $|$✎$| > |(u_1, u_2)| \quad > \quad |(c_1, z_1, c_2, z_2)| = |$✎$|$

Dilithium 3: 8.8 KB 1.6 KB

Two Things You Should Not Do

But We Did in Earlier Versions

🙈 🙉 🙊

Thanks to Thomas Prest and Akira Takahashi for pointing them out to us!

Don't Compress the u-Part

R of degree n
security parameter λ

An idea from Doröz et al. [DHSS20]

Observation: $u \in R_q^k$, but $q^{nk} \gg 2^\lambda$ ($\sim$ finding collisions of random oracle H)

Idea: Compress u via a linear function $T \colon R_q^k \to \mathbb{Z}_q^{n_0}$ such that $q^{n_0} \approx 2^\lambda$ and compute $c = H(T(u), , t)$

Linearity: Necessary for preserving aggregation

Don't Compress the u-Part

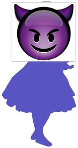
An idea from Doröz et al. [DHSS20]

Observation: $u \in R_q^k$, but $q^{nk} \gg 2^\lambda$ ($\sim$ finding collisions of random oracle H)

Idea: Compress u via a linear function $T \colon R_q^k \to \mathbb{Z}_q^{n_0}$ such that $q^{n_0} \approx 2^\lambda$ and compute $c = H(T(u), 🗎, t)$

Linearity: Necessary for preserving aggregation, **but** allows attacks

Attack: Even against simple signature, given verification key $\mathrm{vk} = t$



Don't Compress the u-Part

R of degree n
security parameter λ

An idea from Doröz et al. [DHSS20]

Observation: $u \in R_q^k$, but $q^{nk} \gg 2^\lambda$ ($\sim$ finding collisions of random oracle H)

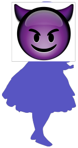
Idea: Compress u via a linear function $T \colon R_q^k \to \mathbb{Z}_q^{n_0}$ such that $q^{n_0} \approx 2^\lambda$ and compute $c = H(T(u), 📄, t)$

Linearity: Necessary for preserving aggregation, **but** allows attacks

Attack: Even against simple signature, given verification key $\mathsf{vk} = t$

Compute $u' = Ay$ and set $c = H(T(u'), 📄, t)$.
Use standard lattice algorithms to find short z such that

$$T(Az) = T(u' + ct) \in \mathbb{Z}_q^{n_0}$$



Set $u := Az - ct$ and output ✏ $= (u, z)$

- z short
- $Az = u + ct$
- $T(u) = T(Az - ct) = T(Az) - T(ct) = T(u')$

Don't Use A Simple Sum

$\mathsf{vk}_1 = t_1$

$\text{🗎}_1 \quad \text{🗎}_2$

$\mathsf{sk}_2 = s_2, \mathsf{vk}_2 = t_2 = As_2$

$u_1 = Ay_1$

$c_1 = H(u_1, \text{🗎}_1, t_1)$

$u_2 = Ay_2 - c_1 t_1$

$c_2 = H(u_2, \text{🗎}_2, t_2)$

$z_2 = s_2 c_2 + y_2$ (rej. sampling)

$\text{✎} = (u_1, u_2, z), \quad z = y_1 + z_2$

Correct forgery:

- z is short (has the right distribution)
- $Az = Ay_1 + Az_2 = u_1 + t_2 c_2 + Ay_2 = u_1 + t_2 c_2 + u_2 + c_1 t_1$

Don't Use A Simple Sum

🖹$_1$ 🖹$_2$

$\mathsf{vk}_1 = t_1$

$\mathsf{sk}_2 = s_2, \mathsf{vk}_2 = t_2 = As_2$

$u_1 = Ay_1$

$c_1 = H(u_1,$ 🖹$_1, t_1)$

$u_2 = Ay_2 - c_1 t_1$

$c_2 = H(u_2,$ 🖹$_2, t_2)$

$z_2 = s_2 c_2 + y_2$ (rej. sampling)

✎ $= (u_1, u_2, z), \quad z = y_1 + z_2$

Correct forgery:

- z is short (has the right distribution)
- $Az = Ay_1 + Az_2 = u_1 + t_2 c_2 + Ay_2 = u_1 + t_2 c_2 + u_2 + c_1 t_1$

Fix:

- Random linear combination of the z_i parts [CGKN21]
- Coefficients from a large enough space ($\{-1, 1\}$ from [DHSS20] not big enough)

Related Works and Open Questions

Related works on lattices

- MMSA(TK) [DHSS20] but unfixed issues!
- Squirrel [FSZ22] in synchronized setting
- Inter-active aggregation (aka multi-signatures) [DOTT21, BTT22]
- Sequential half-aggregation of GPV-signatures [BB14, WW19]

Follow-Up

- Sequential half-aggregation of FSwA-signatures

Related Works and Open Questions

Related works on lattices

- MMSA(TK) [DHSS20] but unfixed issues!
- Squirrel [FSZ22] in synchronized setting
- Inter-active aggregation (aka multi-signatures) [DOTT21, BTT22]
- Sequential half-aggregation of GPV-signatures [BB14, WW19]

Follow-Up

- Sequential half-aggregation of FSwA-signatures

Open questions ?

- Lattice-based signature with
  - public aggregation
  - compactness
  - proof of security in standard setting
- For unbounded number of parties?

Related Works and Open Questions

Related works on lattices

- MMSA(TK) [DHSS20] but unfixed issues!
- Squirrel [FSZ22] in synchronized setting
- Inter-active aggregation (aka multi-signatures) [DOTT21, BTT22]
- Sequential half-aggregation of GPV-signatures [BB14, WW19]

Follow-Up

- Sequential half-aggregation of FSwA-signatures

Open questions ?

- Lattice-based signature with
  - public aggregation
  - compactness
  - proof of security in standard setting
- For unbounded number of parties?

Thank you

Rachid El Bansarkhani and Johannes Buchmann.
Towards lattice based aggregate signatures.
In *AFRICACRYPT*, volume 8469 of *Lecture Notes in Computer Science*, pages 336–355. Springer, 2014.

Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham.
Aggregate and verifiably encrypted signatures from bilinear maps.
In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.

Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi.
Musig-l: Lattice-based multi-signature with single-round online phase, 2022.
Accepted at Crypto 2022.

Konstantinos Chalkias, François Garillot, Yashvanth Kondi, and Valeria Nikolaenko.
Non-interactive half-aggregation of eddsa and variants of schnorr signatures.
In *CT-RSA*, volume 12704 of *Lecture Notes in Computer Science*, pages 577–608. Springer, 2021.

Yarkin Doröz, Jeffrey Hoffstein, Joseph H. Silverman, and Berk Sunar.
MMSAT: A scheme for multimessage multiuser signature aggregation.
IACR Cryptol. ePrint Arch., page 520, 2020.

Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé.
Crystals-dilithium: A lattice-based digital signature scheme.
IACR Trans. Cryptogr. Hardw. Embed. Syst., 2018(1):238–268, 2018.

Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi.
Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices.
In *Public Key Cryptography (1)*, volume 12710 of *Lecture Notes in Computer Science*, pages 99–130. Springer, 2021.

Nils Fleischhacker, Mark Simkin, and Zhenfei Zhang.
Squirrel: Efficient synchronized multi-signatures from lattices.
IACR Cryptol. ePrint Arch., page 694, 2022.

Vadim Lyubashevsky.
Lattice signatures without trapdoors.
In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.

Zhipeng Wang and Qianhong Wu.
A practical lattice-based sequential aggregate signature.

In *ProvSec*, volume 11821 of *Lecture Notes in Computer Science*, pages 94–109. Springer, 2019.