Towards Aggregating Lattice Signatures: Linear Aggregation With Compression

Katharina Boudgoust Adeline Roux-Langlois

Univ Rennes, CNRS, IRISA

Aarhus Crypto Seminar 11 March 2021, Online

Where and who am I?

Image: Open Street Map

Where and who am I?

Lattice-based cryptography

Image: Open Street Map

Where and who am I?

Lattice-based cryptography

& Module Learning With Errors (+ Variants)

Image: Open Street Map

Context & History

The purpose of cryptology:

- **confidentiality**
- **authenticity**
- **integrity**

Context & History

The purpose of cryptology:

- **confidentiality** historically the most important (e.g., WWII)
- **authenticity**
- **integrity**

Context & History

The purpose of cryptology:

- **confidentiality** historically the most important (e.g., WWII)
- **authenticity** increasingly gaining importance (e.g., online tasks)
- **integrity**

Context & History

The purpose of cryptology:

- **confidentiality** historically the most important (e.g., WWII)
- **authenticity** increasingly gaining importance (e.g., online tasks)
- **integrity**

Digital Signatures ensure authenticity and integrity!

1976 First described by Diffie and Hellman [DH76]

1978 First realized by Rivest, Shamir and Adleman [RSA78]

1988 Rigorously defined security notions by Goldwasser, Micali and Rivest [GMR88]

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

message 📄

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

$$📄, ✎ = \mathsf{Sig}(📄, \mathsf{sk}) \longrightarrow$$

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

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

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

message m_1

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

$$m_1, \sigma_1 = \mathsf{Sig}(m_1, \mathsf{sk}_1) \longrightarrow$$

$$\{0, 1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_1, m_1, \sigma_1)$$

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

message m_1

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

$$m_1, \sigma_1 = \mathsf{Sig}(m_1, \mathsf{sk}_1) \longrightarrow$$

message m_2

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

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_1, m_1, \sigma_1)$

$$m_2, \sigma_2 = \mathsf{Sig}(m_2, \mathsf{sk}_2) \longrightarrow$$

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_2, m_2, \sigma_2)$

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

message m_1

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

$m_1, \sigma_1 = \mathsf{Sig}(m_1, \mathsf{sk}_1)$ $\longrightarrow$

message m_2

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

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_1, m_1, \sigma_1)$

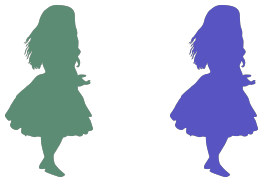
$m_2, \sigma_2 = \mathsf{Sig}(m_2, \mathsf{sk}_2)$ $\longrightarrow$

$\{0,1\} \leftarrow \mathsf{Vf}(\mathsf{vk}_2, m_2, \sigma_2)$

Q: Can we combine both (m_1, σ_1) and (m_2, σ_2) to something shorter?

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

Aggregate Signatures: AggSig and AggVf [BGLS03]



$\sigma_j = \mathsf{Sig}(m_j, \mathsf{sk}_j)$ for $j = 1, 2$

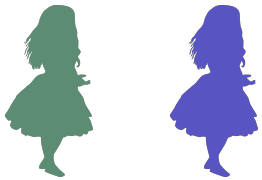
$\mathsf{vk} = (\mathsf{vk}_1, \mathsf{vk}_2)$

$\sigma \leftarrow \mathsf{AggSig}(\mathsf{vk}, m_1, m_2, \sigma_1, \sigma_2)$

$m_1, m_2, \sigma \longrightarrow$

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, m_1, m_2, \sigma)$

Aggregate Signatures: AggSig and AggVf [BGLS03]



$\sigma_j = \mathsf{Sig}(m_j, \mathsf{sk}_j)$ for $j = 1, 2$

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

$\sigma \leftarrow \mathsf{AggSig}(\mathsf{vk}, m_1, m_2, \sigma_1, \sigma_2)$

$\xrightarrow{m_1, m_2, \sigma}$

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, m_1, m_2, \sigma)$

Properties

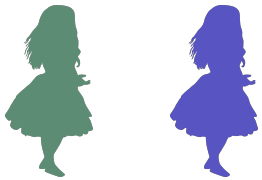
Correctness

Public aggregation

Compactness

Unforgeability

Aggregate Signatures: AggSig and AggVf [BGLS03]



$\sigma_j = \mathsf{Sig}(m_j, \mathsf{sk}_j)$ for $j = 1, 2$

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

$\sigma \leftarrow \mathsf{AggSig}(\mathsf{vk}, m_1, m_2, \sigma_1, \sigma_2)$

only public input

m_1, m_2, σ →

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, m_1, m_2, \sigma)$

Properties

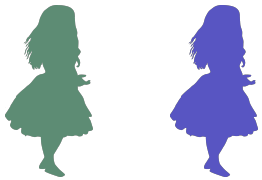
Correctness

Public aggregation

Compactness

Unforgeability

Aggregate Signatures: AggSig and AggVf [BGLS03]



$\sigma_j = \mathsf{Sig}(m_j, \mathsf{sk}_j)$ for $j = 1, 2$

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

$\sigma \leftarrow \mathsf{AggSig}(\mathsf{vk}, m_1, m_2, \sigma_1, \sigma_2)$

only public input

m_1, m_2, σ

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, m_1, m_2, \sigma)$

Properties

Correctness

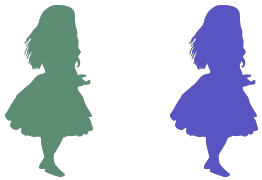
Public aggregation

Compactness

Unforgeability

in chosen-key model (later more)

Aggregate Signatures: AggSig and AggVf [BGLS03]



$\sigma_j = \mathsf{Sig}(m_j, \mathsf{sk}_j)$ for $j = 1, 2$

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

$\sigma \leftarrow \mathsf{AggSig}(\mathsf{vk}, m_1, m_2, \sigma_1, \sigma_2)$

$m_1, m_2, \sigma \longrightarrow$

$\{0, 1\} \leftarrow \mathsf{AggVf}(\mathsf{vk}, m_1, m_2, \sigma)$

Properties

Correctness

Public aggregation

Compactness

Unforgeability

Applications

Consensus Protocols

Certificate Chains

Blockchains

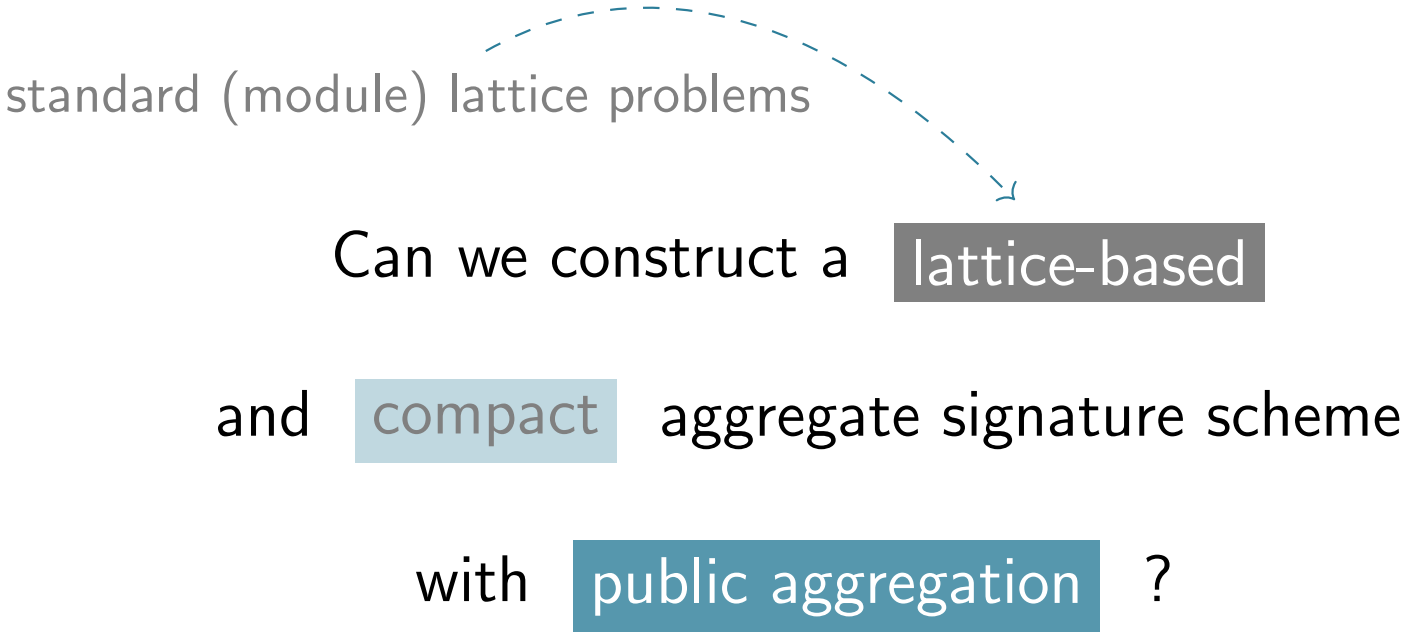
Research Question:

Can we construct a lattice-based

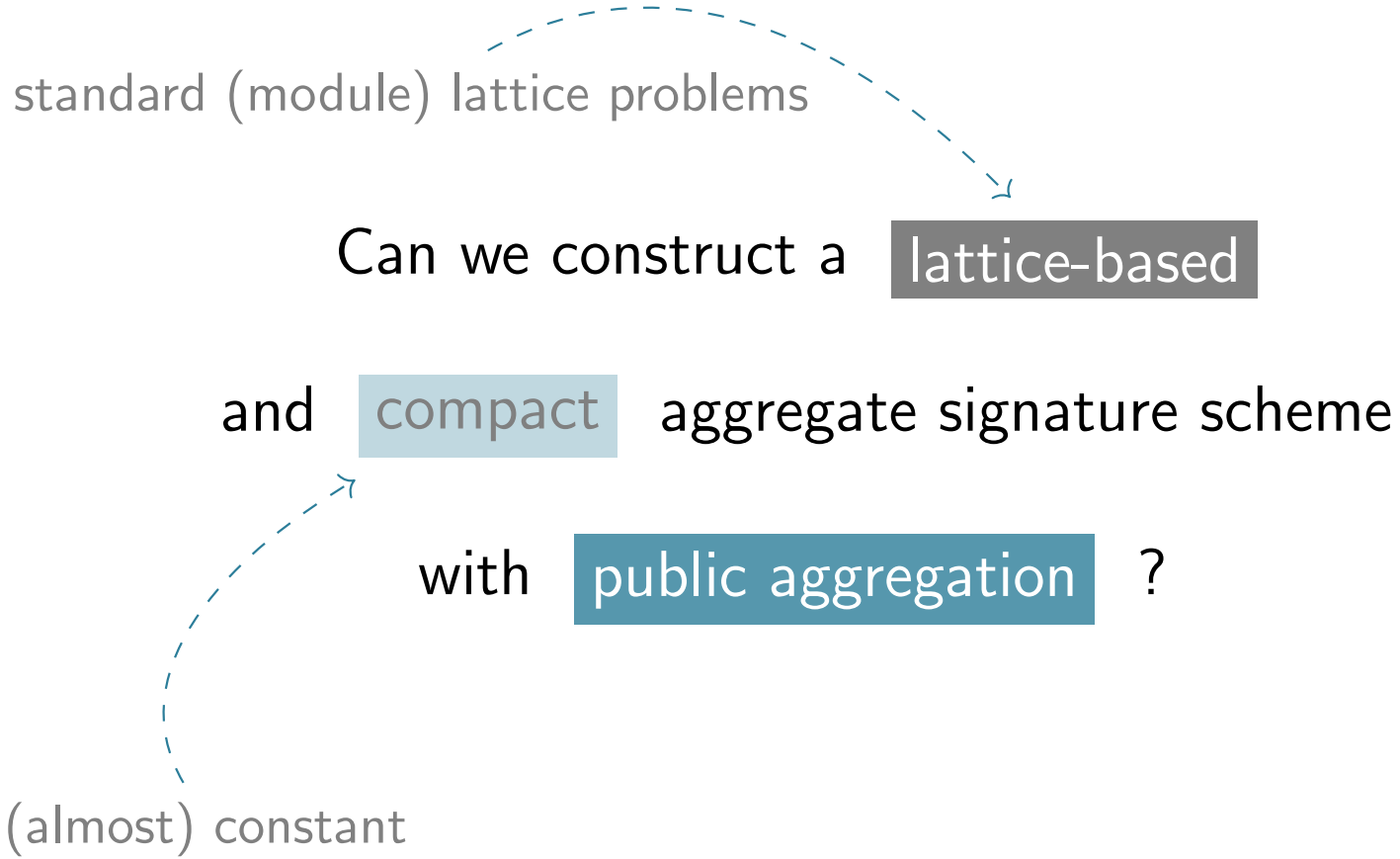
and compact aggregate signature scheme

with public aggregation ?

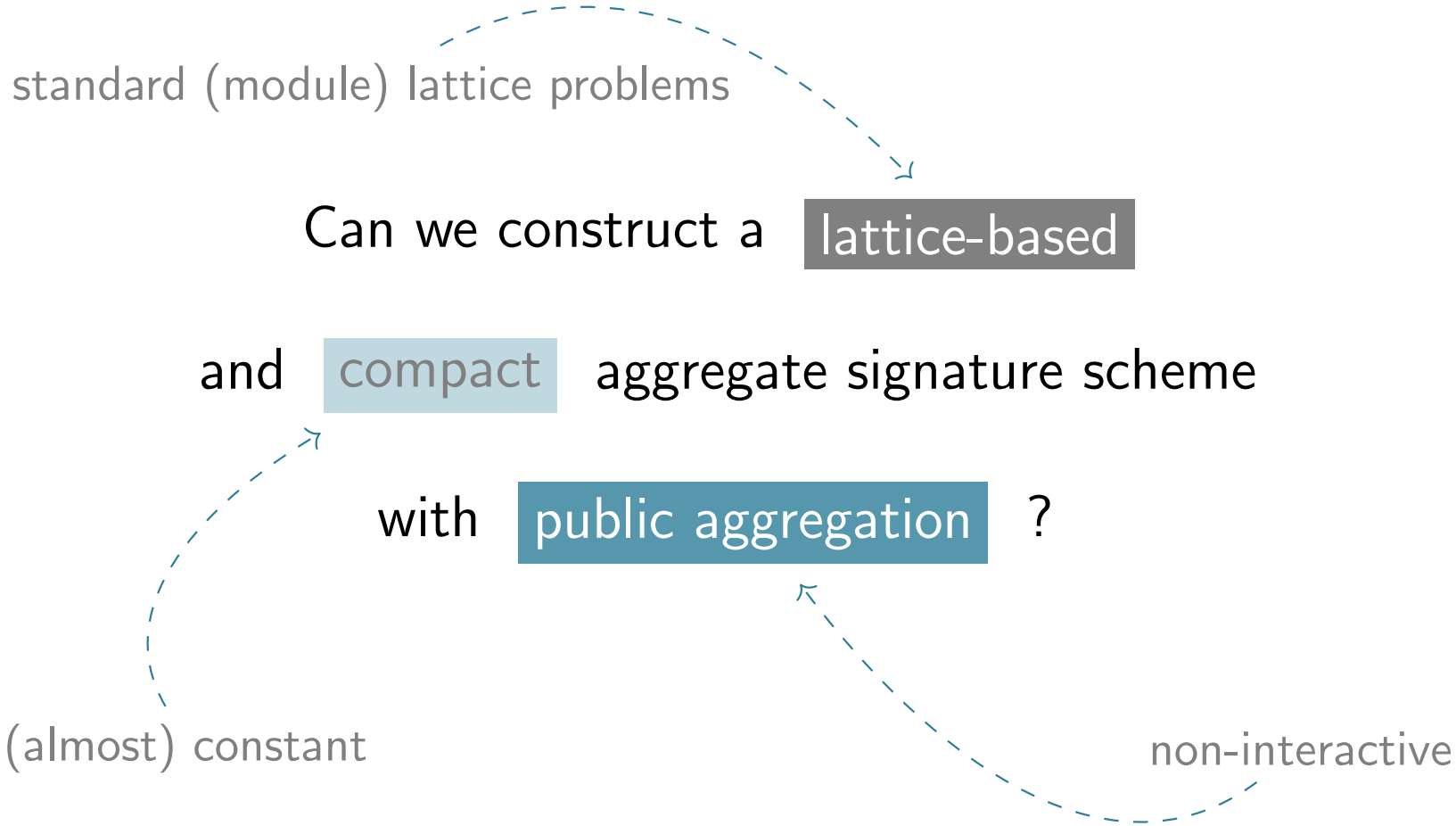
Research Question:



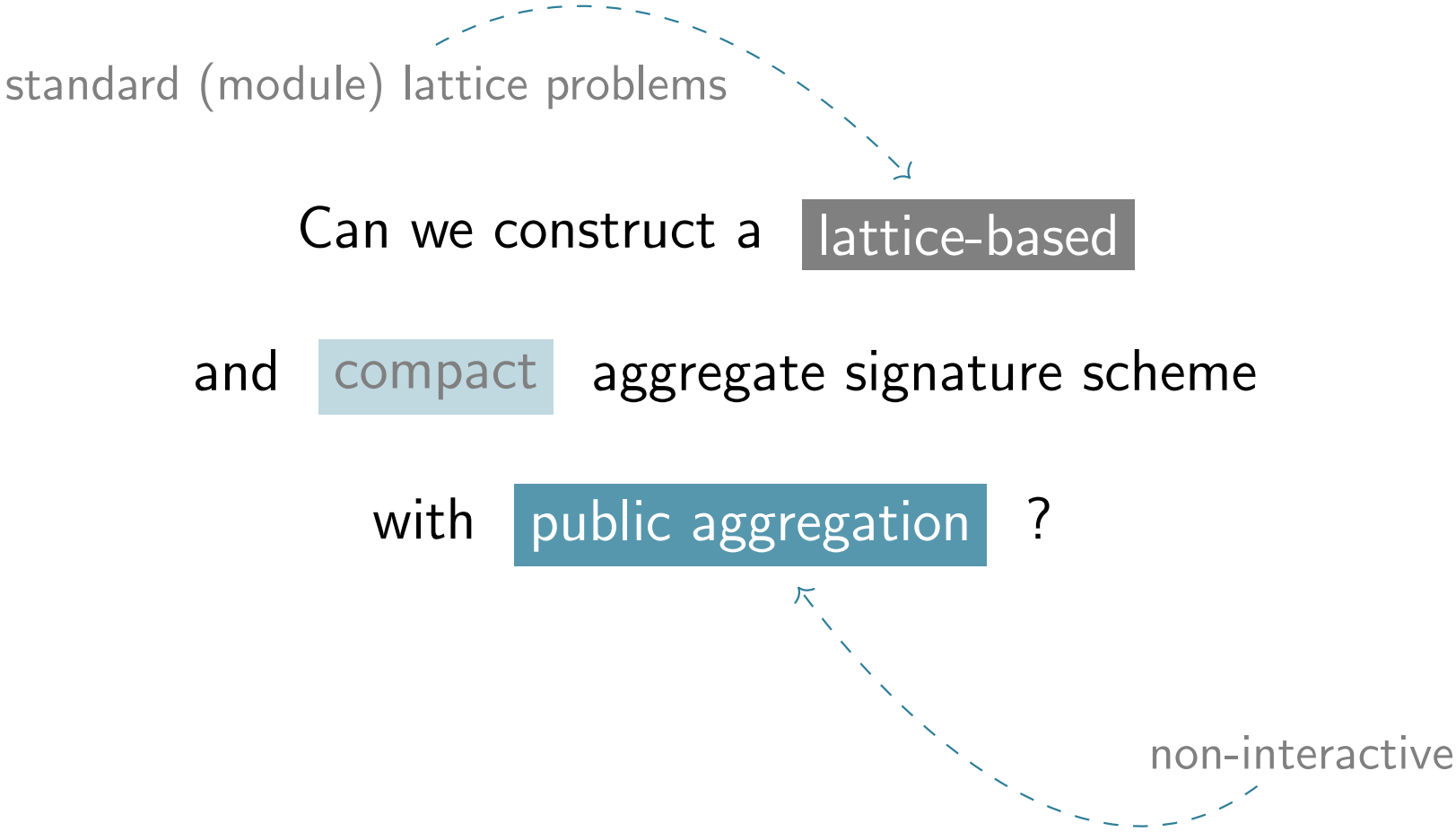
Research Question:



Research Question:



Research Question:



Today: concentrate on lattice-based and public aggregation, see

ia.cr/2021/263

Outline

Outline

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times\ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times\ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

n power of 2, q prime

k and ℓ small constants

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times\ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

message

KGen :

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

$\mathsf{vk} = t = As$

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times \ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

message

KGen :

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

$\mathsf{vk} = t = As$

Sig :

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

$c = H_c(u, \text{message}) \in R$ small

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

$\xrightarrow{\text{message}, = (c, z)}$

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times \ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

message

KGen :

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

$\mathsf{vk} = t = As$

Sig :

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

$c = H_c(u, \text{msg}) \in R$ small

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

$\xrightarrow{\text{msg}, \text{sig} = (c, z)}$

Vf :

if $c = H_c(Az - tc, \text{msg})$

and z small, accept

Fiat-Shamir with Aborts (FSwA) [Lyu09]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $A' \leftarrow U(R_q^{k\times \ell})$ defining $A = [A'|I_k]$ be public parameters and $H_c : \{0,1\}^* \to C$ be a random oracle

message

KGen :

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

$\mathsf{vk} = t = As$

Sig :

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

$c = H_c(u, \text{message})$

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

$\xrightarrow{\text{message}, \text{signature} = (c, z)}$

$$\begin{aligned} & Az - tc \\ = \; & A(sc + y) - (As)c \\ = \; & Ay \\ = \; & u \end{aligned}$$

Vf :

if $c = H_c(Az - tc, \text{message})$

and z small, accept

Fiat-Shamir with Aborts (minor modifications)

Let R_q and A be public and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be a linear function, $n_0 \cdot \log_2(q) \approx 2\lambda$.

message

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

$\mathsf{vk} = t = As$

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

$c = H_c($ T (u) , , t) small

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

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

compute $c = H_c($ T $(u), , t)$

if $Az - tc = u$ and z small

accept

Fiat-Shamir with Aborts (minor modifications)

Let R_q and A be public and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be a linear function, $n_0 \cdot \log_2(q) \approx 2\lambda$.

message

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

$\mathsf{vk} = t = As$

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

$c = H_c(\ T\ (\mathrm{u})\ ,$, $\mathrm{t}\ )$ small

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

efficiency

$|T(u)| \approx 2\lambda = 256$
$|u| = nk \log_2 q \approx 40000$

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

compute $c = H_c(\ T\ (u), , t)$

if $Az - tc = u$ and z small

accept

Fiat-Shamir with Aborts (minor modifications)

Let R_q and A be public and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be a linear function, $n_0 \cdot \log_2(q) \approx 2\lambda$.

message

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

$\mathsf{vk} = t = As$

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

$c = H_c(\ T\ (\mathrm{u})\ , \text{msg},\ \mathrm{t}\ )$ small

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

security

efficiency

$|T(u)| \approx 2\lambda = 256$

$|u| = nk \log_2 q \approx 40000$

msg, sig $= (\ u\ , z)$

compute $c = H_c(\ T\ (u), \text{msg}, t)$

if $Az - tc = u$ and z small

accept

Fiat-Shamir with Aborts (minor modifications)

Let R_q and A be public and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be a linear function, $n_0 \cdot \log_2(q) \approx 2\lambda$.

message

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

$\mathsf{vk} = t = As$

security

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

$c = H_c($ T (u) , , t) small

functionality

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

, = (u , z)

efficiency

$|T(u)| \approx 2\lambda = 256$
$|u| = nk \log_2 q \approx 40000$

compute $c = H_c($ T $(u), , t)$

if $Az - tc = u$ and z small

accept

Naive Aggregation for FSwA Signature

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle

message doc_1

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

$y_1, u_1, c_1 = H_c(T(u_1), \text{doc}_1, t_1)$

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

$\sigma_1 = (u_1, z_1)$

message doc_2

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

$y_2, u_2, c_2 = H_c(T(u_2), \text{doc}_2, t_2)$

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

$\sigma_2 = (u_2, z_2)$

Naive Aggregation for FSwA Signature

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle

message doc_1

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

$y_1, u_1, c_1 = H_c(T(u_1), \text{doc}_1, t_1)$

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

$\sigma_1 = (u_1, z_1)$

message doc_2

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

$y_2, u_2, c_2 = H_c(T(u_2), \text{doc}_2, t_2)$

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

$\sigma_2 = (u_2, z_2)$

Naive idea: $\sigma = (u = u_1 + u_2, z = z_1 + z_2) \Rightarrow Az = t_1 c_1 + t_2 c_2 + u$

Naive Aggregation for FSwA Signature

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle

message doc_1

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

$y_1, u_1, c_1 = H_c(T(u_1), \text{doc}_1, t_1)$

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

$\text{pen}_1 = (u_1, z_1)$

message doc_2

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

$y_2, u_2, c_2 = H_c(T(u_2), \text{doc}_2, t_2)$

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

$\text{pen}_2 = (u_2, z_2)$

💡 Naive idea: $\text{pen} = (u = u_1 + u_2, z = z_1 + z_2) \Rightarrow Az = t_1 c_1 + t_2 c_2 + u$

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

Naive Aggregation for FSwA Signature

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle

message m_1

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

$y_1, u_1, c_1 = H_c(T(u_1), m_1, t_1)$

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

$\sigma_1 = (u_1, z_1)$

message m_2

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

$y_2, u_2, c_2 = H_c(T(u_2), m_2, t_2)$

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

$\sigma_2 = (u_2, z_2)$

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

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

⚙ Inter-active solution: agree on the same $u_1 = u_2$

⚙ Alternative: provide enough information by including all $T(u_j)$
(compression needed, see [DHSS20] and our scheme)

Outline

Linear Aggregation With Compression

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle.

Given N signatures $\Sigma = (✎_j) = (u_j, z_j)_{j \in [N]}$ for verification keys $\mathsf{VK} = (t_j)_j$ on the messages $M = (🗎_j)_j$.

Linear Aggregation With Compression

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle.

Given N signatures $\Sigma = (✎_j) = (u_j, z_j)_{j\in[N]}$ for verification keys $\mathsf{VK} = (t_j)_j$ on the messages $M = (🗎_j)_j$.

$\mathsf{AggSig}(\mathsf{VK}, M, \Sigma)$: Compute $T(u_j) \quad \forall j \in [N]$
set $z = \sum_j z_j \in R_q^{\ell+k}$
and $u = \sum_j u_j \in R_q^k$
if $\|z\|_2$ small, return $✎ = (u, (T(u_j))_j, z)$;
else return $\perp$;

$\mathsf{AggVf}(\mathsf{VK}, M, ✎)$: Query $c_j = H_c(T(u_j), t_j, m_j) \quad \forall j \in [N]$
If $\|z\|_2$ small, and if $T(u) = \sum_j T(u_j)$
and if $A \cdot z = \sum_j (t_j \cdot c_j) + u$
return 1; else return 0;

Linear Aggregation With Compression

Let R_q, A and $T : R_q^k \to \mathbb{Z}_q^{n_0}$ be public, H_c random oracle.

Given N signatures $\Sigma = (✒_j) = (u_j, z_j)_{j\in[N]}$ for verification keys $\mathsf{VK} = (t_j)_j$ on the messages $M = (🗎_j)_j$.

$\mathsf{AggSig}(\mathsf{VK}, M, \Sigma)$: Compute $T(u_j) \quad \forall j \in [N]$
set $z = \sum_j z_j \in R_q^{\ell+k}$
and $u = \sum_j u_j \in R_q^k$
if $\|z\|_2$ small, return $✒ = (u, (T(u_j))_j, z)$;
else return $\perp$;

$\mathsf{AggVf}(\mathsf{VK}, M, ✒)$: Query $c_j = H_c(T(u_j), t_j, m_j) \quad \forall j \in [N]$
If $\|z\|_2$ small, and if $T(u) = \sum_j T(u_j)$
and if $A \cdot z = \sum_j (t_j \cdot c_j) + u$
return 1; else return 0;

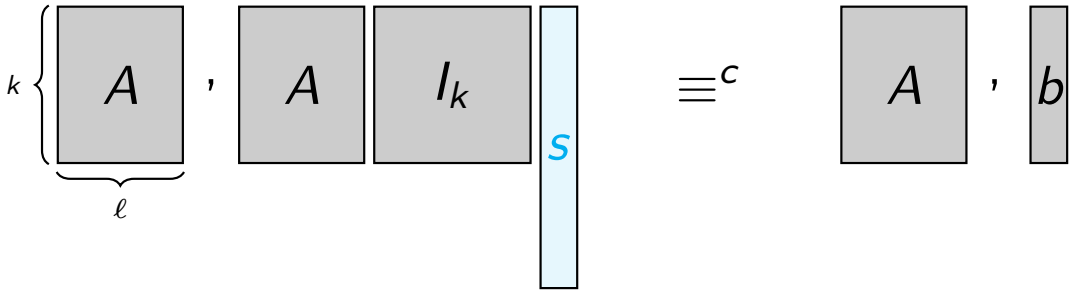
Correctness: Linearity of matrix-vector multiplication
Compactness: $|✒| \ll |\Sigma|$ $N = 10^3$ and Dilithium III: 43.7 KB vs. 2701 KB
Smallness: From rejection sampling $z_j \sim D_\alpha^{\ell+k}$, then $z \sim D_{\sqrt{N}\alpha}^{\ell+k}$

Overview

Hard Problems Over Module Lattices [LS15, BJRW20]

Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $k, \ell \in \mathbb{N}$

Module Learning With Errors (Module-LWE): Distinguish

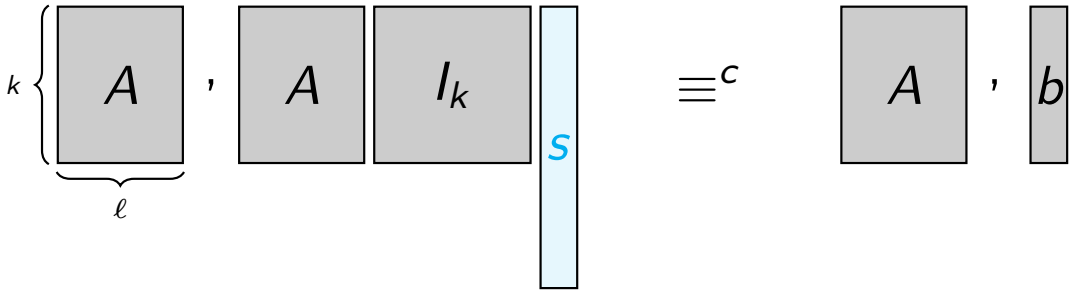


where $s \in R^{\ell+k}$ is of small norm and $(A, b) \leftarrow U(R_q^{k\times\ell}) \times U(R_q^k)$.

Hard Problems Over Module Lattices [LS15, BJRW20]

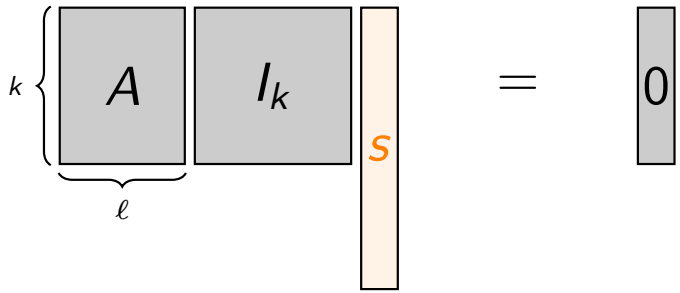
Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $k, \ell \in \mathbb{N}$

Module Learning With Errors (Module-LWE): Distinguish



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

Module Short Integer Solution (Module-SIS): Given $A \leftarrow U(R_q^{k\times\ell})$, find s s.t.



where $s \in R_q^{\ell+k} \setminus \{0\}$ is of small norm.

Chosen-Key Model from [BGLS03]

$\Pi_{AS} = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf}, \mathsf{AggSig}, \mathsf{AggVf})$ aggregate signature scheme

$N \in \mathbb{N}$ number of signatures to aggregate

Challenger $\mathcal{B}$ | Adversary $\mathcal{A}$

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

$\mathsf{vk}_N \longrightarrow$

Chosen-Key Model from [BGLS03]

$\Pi_{AS} = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf}, \mathsf{AggSig}, \mathsf{AggVf})$ aggregate signature scheme

$N \in \mathbb{N}$ number of signatures to aggregate

Challenger $\mathcal{B}$ | Adversary $\mathcal{A}$

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

vk_N →

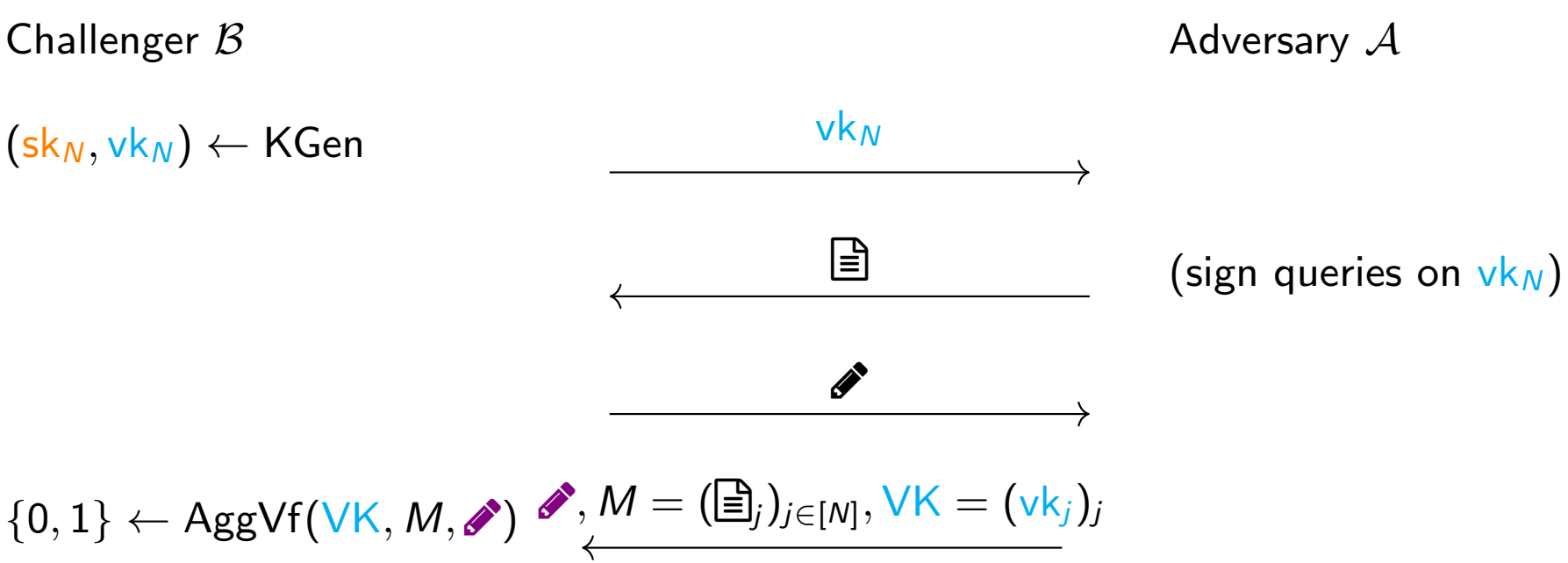
← (sign queries on vk_N)

→

Chosen-Key Model from [BGLS03]

$\Pi_{AS} = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf}, \mathsf{AggSig}, \mathsf{AggVf})$ aggregate signature scheme

$N \in \mathbb{N}$ number of signatures to aggregate



$\mathcal{A}$ wins the game if $1 \leftarrow \mathsf{AggVf}$ and m_N not queried before

Π_{AS} secure against existential forgery in chosen-key model if success proba of any PPT $\mathcal{A}$ is negligibly small

Security Proof: Statement

Random Oracle $H_c : \{0,1\}^* \to C$

Theorem (Security)

Assume the hardness of Module-LWE *and of* Module-SIS. *Then the aggregate signature Π_{AS} presented before is secure against existential forgery in the aggregate chosen-key model in the ROM. The advantage of some PPT adversary $\mathcal{A}$ against Π_{AS} is bounded above by*

$$\mathsf{AdvAggSig}_{\mathcal{A}} \leq \mathsf{Adv}_{\text{Module-LWE}} + N_q / |C| + \sqrt{N_q \cdot \mathsf{Adv}_{\text{Module-SIS}}} + \mathsf{negl}(n),$$

where $\mathcal{A}$ makes at most N_{H_c} queries to H_c and at most N_{Sig} queries to the signing oracle and $N_q = N_{H_c} + N_{\mathsf{Sig}}$.

Security Proof: Highlevel as in [DOTT20]

Game 0: Original security game:
challenger honestly generates $(\mathsf{sk}_N, \mathsf{vk}_N)$
and honestly answers signing queries

Game 1: Challenger simulates signing procedure
without using sk_N, only vk_N

Game 2 Challenger generates a lossy key $\mathsf{vk}_N \leftarrow U(R_q^k)$

Security Proof: Highlevel as in [DOTT20]

Game 0: Original security game:
challenger honestly generates $(\mathsf{sk}_N, \mathsf{vk}_N)$
and honestly answers signing queries

statistically close due to rejection sampling

Game 1: Challenger simulates signing procedure
without using sk_N, only vk_N

Game 2 Challenger generates a lossy key $\mathsf{vk}_N \leftarrow U(R_q^k)$

Security Proof: Highlevel as in [DOTT20]

Game 0: Original security game:
challenger honestly generates $(\mathsf{sk}_N, \mathsf{vk}_N)$
and honestly answers signing queries

Game 1: Challenger simulates signing procedure
without using sk_N, only vk_N

Game 2 Challenger generates a lossy key $\mathsf{vk}_N \leftarrow U(R_q^k)$

statistically close due to rejection sampling

computationally close assuming Module-LWE

Security Proof: Highlevel as in [DOTT20]

Game 0: Original security game:
challenger honestly generates $(\mathsf{sk}_N, \mathsf{vk}_N)$
and honestly answers signing queries

statistically close due to rejection sampling

Game 1: Challenger simulates signing procedure
without using sk_N, only vk_N

computationally close assuming Module-LWE

Game 2 Challenger generates a lossy key $\mathsf{vk}_N \leftarrow U(R_q^k)$

In Game 2: rewind challenger and adversary (apply General Forking Lemma) to obtain two different forgeries:

$$✎ = (u, (T(u_j))_j, z) \text{ and } ✎' = (u', (T(u_j)')_j, z'),$$

where $u = u'$, $T(u_j) = T(u_j)' \quad \forall j \in [N]$ and $c_j = c_j' \quad \forall j \in [N-1]$, but $c_N \neq c_N'$.

$$Az - t_N c_N = u + \sum_{j \in [N-1]} t_j c_j = u' + \sum_{j \in [N-1]} t_j c_j' = Az' - t_N c_N'$$

leads to solution to Module-SIS for matrix $[A|t_N]$.

Overview

Further work and open questions

Related work

- Inter-active aggregate signatures (aka multi-signatures)

Work in progress

- Realize relaxed aggregation: sequential aggregate signature
  aggregation follows a sequential order

Open questions **?**

- (almost) constant size and public aggregation
- Maybe tighter security proof using Abdalla et al. [AFLT16]
- Security proof in **quantum** random oracle model

Further work and open questions

Related work

- Inter-active aggregate signatures (aka multi-signatures)

Work in progress

- Realize relaxed aggregation: sequential aggregate signature
  aggregation follows a sequential order

Open questions **?**

- (almost) constant size and public aggregation
- Maybe tighter security proof using Abdalla et al. [AFLT16]
- Security proof in **quantum** random oracle model

Thank you.


M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi.
Tightly secure signatures from lossy identification schemes.
J. Cryptol., 29(3):597–631, 2016.


D. Boneh, C. Gentry, B. Lynn, and H. Shacham.
Aggregate and verifiably encrypted signatures from bilinear maps.
In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer, 2003.


K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen.
Towards classical hardness of module-lwe: The linear rank case.
In *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 289–317. Springer, 2020.


W. Diffie and M. E. Hellman.
New directions in cryptography.
IEEE Trans. Inf. Theory, 22(6):644–654, 1976.


Y. Doröz, J. Hoffstein, J. H. Silverman, and B. Sunar.
MMSAT: A scheme for multimessage multiuser signature aggregation.
IACR Cryptol. ePrint Arch., 2020:520, 2020.


I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi.
Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices.
IACR Cryptol. ePrint Arch. to appear at PKC 2021, 2020:1110, 2020.


S. Goldwasser, S. Micali, and R. L. Rivest.
A digital signature scheme secure against adaptive chosen-message attacks.
SIAM J. Comput., 17(2):281–308, 1988.


A. Langlois and D. Stehlé.
Worst-case to average-case reductions for module lattices.
Des. Codes Cryptogr., 75(3):565–599, 2015.


V. Lyubashevsky.
Fiat-shamir with aborts: Applications to lattice and factoring-based signatures.
In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information*

Security, Tokyo, Japan, December 6-10, 2009. Proceedings, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

R. L. Rivest, A. Shamir, and L. M. Adleman.
A method for obtaining digital signatures and public-key cryptosystems.
Commun. ACM, 21(2):120–126, 1978.