# MuSig-L

## Lattice-based Multi-Signature with Single-Round Online Phase

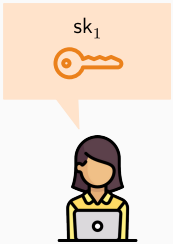
CRYPTO 2022

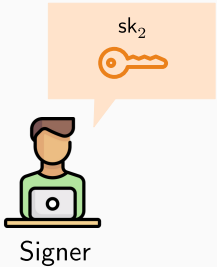
Cecilia Boschini[1] Akira Takahashi[2] Mehdi Tibouchi[3]

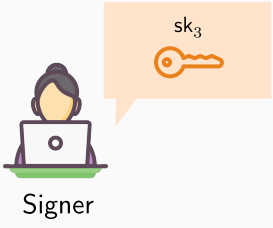
[1]Technion and Reichman University, Israel
[2]Aarhus University, Denmark
[3]NTT Corporation, Japan

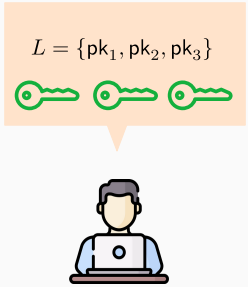



# Interactive Multi-Signature

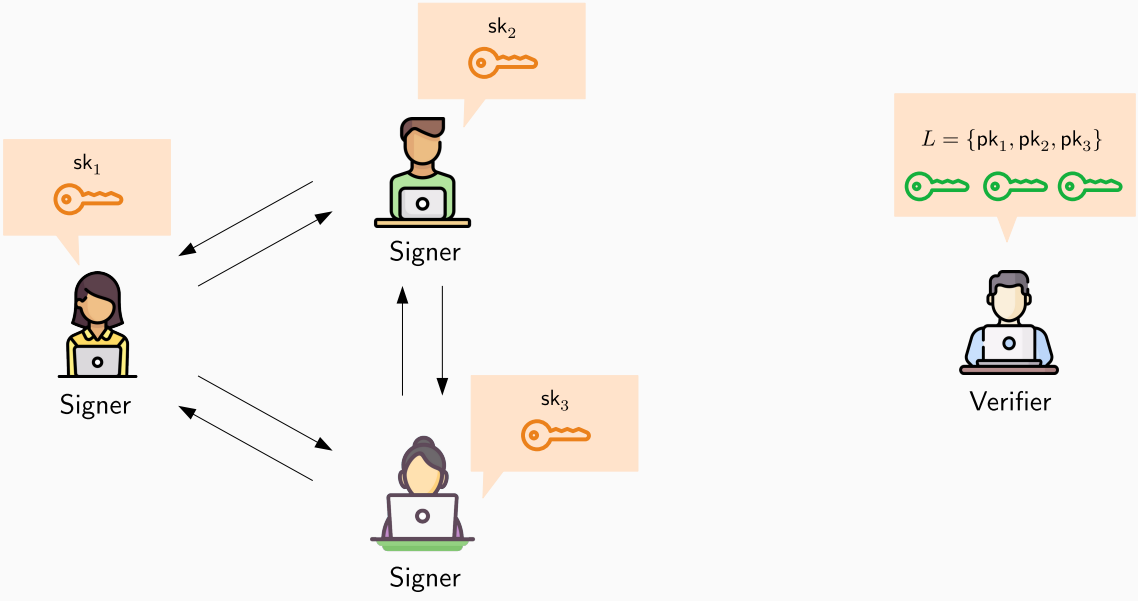


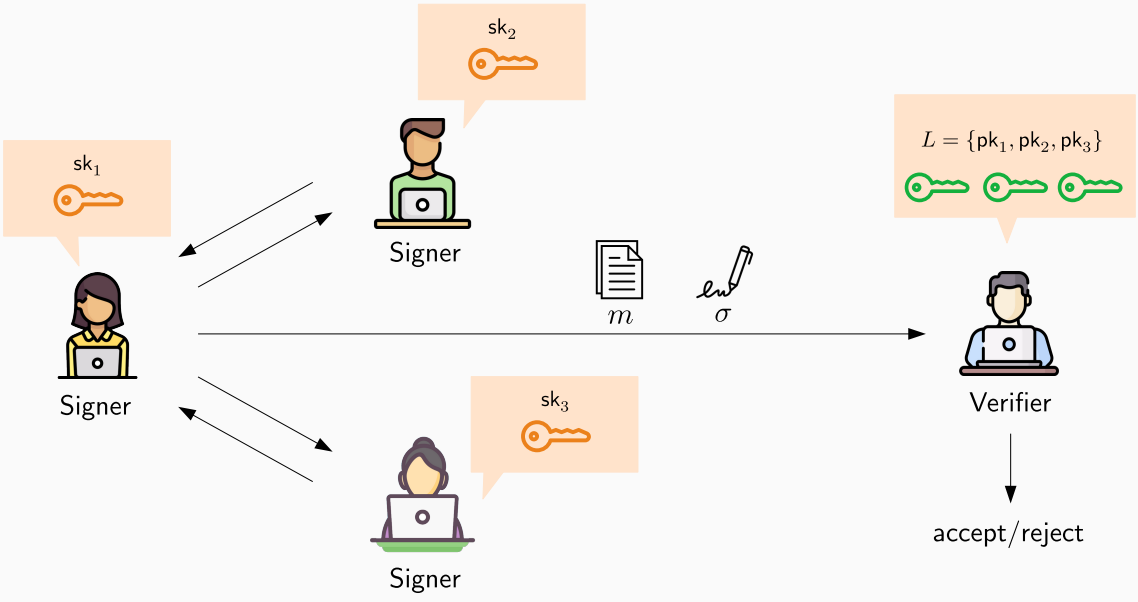



# Interactive Multi-Signature



# Interactive Multi-Signature

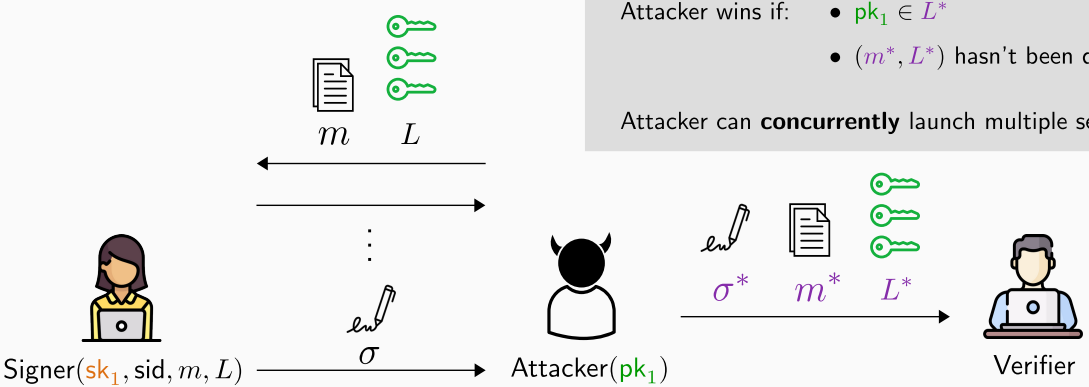


# Security in the Plain Public Key Model

MS-UF-CMA game

Attacker wins if:
- $(\sigma^*, m^*, L^*)$ is valid
- $\mathsf{pk}_1 \in L^*$
- $(m^*, L^*)$ hasn't been queried

Attacker can **concurrently** launch multiple sessions



## Schnorr ID

$r \leftarrow_{\$} \mathbb{Z}_q$

$U := r \cdot G$ $\xrightarrow{U}$

$\xleftarrow{c}$ $c \leftarrow_{\$} \mathbb{Z}_q$

$z := c \cdot \mathsf{sk} + r$ $\xrightarrow{z}$ Accept iff $z \cdot G = c \cdot \mathsf{pk} + U$

Prover($\mathsf{sk}$)

Verifier($\mathsf{pk} = \mathsf{sk} \cdot G$)

# Lattice-based Schnorr

## Schnorr ID

$r \leftarrow_{\$} \mathbb{Z}_q$

$U := r \cdot G$ — $U$ → — $c \leftarrow_{\$} \mathbb{Z}_q$ — $c$ ←

$z := c \cdot \mathsf{sk} + r$ — $z$ → — Accept iff $z \cdot G = c \cdot \mathsf{pk} + U$

Prover($\mathsf{sk}$)

Verifier($\mathsf{pk} = \mathsf{sk} \cdot G$)

## Fiat-Shamir with Aborts ID [Lyu12]

$\mathbf{r} \leftarrow D$

$\mathbf{u} := \mathbf{Ar}$ — $\mathbf{u}$ → — $c \leftarrow_{\$} C \subset R$ — $c$ ←

$\mathbf{z} := c \cdot \mathsf{sk} + \mathbf{r}$

If RejSamp($\mathbf{z}$) $= 0$:

Abort — $\mathbf{z}$ → — Accept iff $\mathbf{Az} = c \cdot \mathsf{pk} + \mathbf{u} \wedge \|\mathbf{z}\| \leq B$

Prover($\mathsf{sk}$)

Verifier($\mathsf{pk} = \mathbf{A} \cdot \mathsf{sk}$)

# Naive Two-round Protocol with Passive Security (DLog)

**Sign($\mathsf{sk}_1, m$)**

$r_1 \leftarrow_\$ \mathbb{Z}_q$

$U_1 := r_1 \cdot G$

$U := U_1 + U_2$

$c := \mathsf{H}(U, m, \mathsf{pk})$

$z_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + r_1$

$z := z_1 + z_2$

$U_1$ →

← $U_2$

$z_1$ →

← $z_2$

**Sign($\mathsf{sk}_2, m$)**

- Key pair: $\mathsf{pk}_i = \mathsf{sk}_i \cdot G$
- Public key aggregation [MPSW19]:
  - $a_i = \mathsf{H}(\{\mathsf{pk}_1, \mathsf{pk}_2\}, \mathsf{pk}_i) \in \mathbb{Z}_q$
  - $\mathsf{pk} := a_1 \cdot \mathsf{pk}_1 + a_2 \cdot \mathsf{pk}_2$
- Works thanks to homomorphism of $f(x) = x \cdot G$

$m, (U, z)$ →

$c := \mathsf{H}(U, m, \mathsf{pk})$

Accept iff

$z \cdot G = c \cdot \mathsf{pk} + U$

Signer$_1$

Signer$_2$

Verifier

# Naive Two-round Protocol with Passive Security (Lattice)

$\mathsf{Sign}(\mathsf{sk}_1, m)$

$\mathbf{r}_1 \leftarrow D_\sigma$

$\mathbf{u}_1 := \mathbf{A}\mathbf{r}_1$

$\mathbf{u} := \mathbf{u}_1 + \mathbf{u}_2$

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$

If $\mathsf{RejSamp}(\mathbf{z}_1) = 0$:

Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1$ →

← $\mathbf{u}_2$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

$\mathsf{Sign}(\mathsf{sk}_2, m)$

- Key pair: $\mathsf{pk}_i = \mathbf{A} \cdot \mathsf{sk}_i \bmod q$
- Public key aggregation:
  - $a_i = \mathsf{H}(\{\mathsf{pk}_1, \mathsf{pk}_2\}, \mathsf{pk}_i) \in C$
  - $\mathsf{pk} := a_1 \cdot \mathsf{pk}_1 + a_2 \cdot \mathsf{pk}_2$
- Works thanks to homomorphism of $f(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$

$m, (\mathbf{u}, \mathbf{z})$ →

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

Accept iff

$\mathbf{A}\mathbf{z} = c \cdot \mathsf{pk} + \mathbf{u}$

$\wedge \|\mathbf{z}\| \leq \sqrt{2} \cdot B$

Signer$_1$

Signer$_2$

Verifier

# Naive Two-round Protocol with Passive Security (Lattice)

$\mathsf{Sign}(\mathsf{sk}_1, m)$

$\mathbf{r}_1 \leftarrow D_\sigma$

$\mathbf{u}_1 := \mathbf{A}\mathbf{r}_1$

$\mathbf{u} := \mathbf{u}_1 + \mathbf{u}_2$

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$

If $\mathsf{RejSamp}(\mathbf{z}_1) = 0$:

Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1$ →

← $\mathbf{u}_2$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

$\mathsf{Sign}(\mathsf{sk}_2, m)$

Signer$_1$

Signer$_2$

- Use Gaussian $D_\sigma$ to benefit from convolution:
  - Given $\mathbf{z}_1, \mathbf{z}_2 \sim D_\sigma$, $\mathbf{z}_1 + \mathbf{z}_2 \sim D_{\sqrt{2}\cdot\sigma}$
- Increase $\sigma$ or parallel repetitions
  - $\rightsquigarrow$ Pick an instance where all signers pass

$m, (\mathbf{u}, \mathbf{z})$ →

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

Accept iff

$\mathbf{A}\mathbf{z} = c \cdot \mathsf{pk} + \mathbf{u}$

$\wedge \|\mathbf{z}\| \leq \sqrt{2} \cdot B$

Verifier

# Insecurity of the Naive Two-round Protocol

Sign($\mathsf{sk}_1, m$)

$\mathbf{r}_1 \leftarrow D_\sigma$

$\mathbf{u}_1 := \mathbf{A}\mathbf{r}_1$

$\mathbf{u} := \mathbf{u}_1 + \mathbf{u}_2$

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$

If $\mathsf{RejSamp}(\mathbf{z}_1) = 0$:

Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1$ →

← $\mathbf{u}_2$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

Sign($\mathsf{sk}_2, m$)

Signer$_1$

Attacker($\mathsf{pk}_1$)

- Simple two-round protocol is insecure against **malicious attackers!**
- **Concurrent** forgery attacks exist:
  - $k$-list sum problem [DEFKLNS19]
  - ROS problem [BLLOR21]

$m, (\mathbf{u}, \mathbf{z})$ →

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

Accept iff

$\mathbf{A}\mathbf{z} = c \cdot \mathsf{pk} + \mathbf{u}$

$\wedge \|\mathbf{z}\| \leq \sqrt{2} \cdot B$

Verifier

# How to Protect against Malicious Attackers in the DLog Setting

## 1. Commit&Open

- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- Reveal $U_1$ only after receiving $\mathsf{C}_2$
- ☹ Requires more rounds

## 2. Trapdoor-Hom-Com

- Generate ck from $m$
- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- $c := \mathsf{H}(\mathsf{C}_1 + \mathsf{C}_2, m, \mathsf{pk})$
- Simulator can equivocate $\mathsf{C}_1$ to anything
- ☺ Preserves round complexity
- ☹ Two-round online phase

## 3. Linear Combinations

- Exchange multiple $U_i^{(j)}$
- Take random linear combinations

$$U := \sum_j b^{(j)} \left( \sum_i U_i^{(j)} \right)$$

- ☺ Single-round online phase!

# How to Protect against Malicious Attackers in the DLog Setting

## 1. Commit&Open

- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- Reveal $U_1$ only after receiving $\mathsf{C}_2$
- ☹ Requires more rounds

## 2. Trapdoor-Hom-Com

- Generate ck from $m$
- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- $c := \mathsf{H}(\mathsf{C}_1 + \mathsf{C}_2, m, \mathsf{pk})$
- Simulator can equivocate $\mathsf{C}_1$ to anything
- ☺ Preserves round complexity
- ☹ Two-round online phase

## 3. Linear Combinations

- Exchange multiple $U_i^{(j)}$
- Take random linear combinations

$$U := \sum_j b^{(j)} \left( \sum_i U_i^{(j)} \right)$$

- ☺ Single-round online phase!

# How to Protect against Malicious Attackers in the DLog Setting

## 1. Commit&Open

- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- Reveal $U_1$ only after receiving $\mathsf{C}_2$
- ☹ Requires more rounds

## 2. Trapdoor-Hom-Com

- Generate ck from $m$
- Send $\mathsf{C}_1 = \mathsf{Com}_{\mathsf{ck}}(U_1)$
- $c := \mathsf{H}(\mathsf{C}_1 + \mathsf{C}_2, m, \mathsf{pk})$
- Simulator can equivocate $\mathsf{C}_1$ to anything
- ☺ Preserves round complexity
- ☹ Two-round online phase

## 3. Linear Combinations

- Exchange multiple $U_i^{(j)}$
- Take random linear combinations

$$U := \sum_j b^{(j)} \left( \sum_i U_i^{(j)} \right)$$

- ☺ Single-round online phase!

# Landscape of Schnorr-like Multi-Signatures

| # Round | Method | DLog | Lattice |
|---|---|---|---|
| 3 | Commit&Open | BN06, MuSig | ES16,MJ19,FH20,BK20 |
| 2 | TD-Hom-Com | mBCJ, HBMS | DOTT21 |
| 1 (Off) + 1 (On) | Linear Combination | MuSig2, DWMS | MuSig-L |

- "Usual" Schnorr–FSwA translation: **DLog** $\mapsto$ **SIS**
- MuSig2 and DWMS rely on the **AGM** or (algebraic) **"one-more" DLog**

*Q. Can we construct a scheme with single-round online phase from* ***standard*** *(module) LWE and SIS assumptions?*

# Landscape of Schnorr-like Multi-Signatures

| # Round | Method | DLog | Lattice |
|---|---|---|---|
| 3 | Commit&Open | BN06, MuSig | ES16,MJ19,FH20,BK20 |
| 2 | TD-Hom-Com | mBCJ, HBMS | DOTT21 |
| 1 (Off) + 1 (On) | Linear Combination | MuSig2, DWMS | MuSig-L |

- "Usual" Schnorr–FSwA translation: **DLog** $\mapsto$ **SIS**
- MuSig2 and DWMS rely on the **AGM** or (algebraic) **"one-more" DLog**

*Q. Can we construct a scheme with single-round online phase from* ***standard*** *(module) LWE and SIS assumptions?*

# Landscape of Schnorr-like Multi-Signatures

| # Round | Method | DLog | Lattice |
|---|---|---|---|
| 3 | Commit&Open | BN06, MuSig | ES16,MJ19,FH20,BK20 |
| 2 | TD-Hom-Com | mBCJ, HBMS | DOTT21 |
| 1 (Off) + 1 (On) | Linear Combination | MuSig2, DWMS | MuSig-L |

- "Usual" Schnorr–FSwA translation: **DLog** $\mapsto$ **SIS**
- MuSig2 and DWMS rely on the **AGM** or (algebraic) **"one-more" DLog**

*Q. Can we construct a scheme with single-round online phase from* ***standard*** *(module) LWE and SIS assumptions?*

# MuSig-L

**Sign($\mathsf{sk}_1, m$)**

For $j = 1, \ldots, \ell$:
$\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
$\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$
If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

**Sign($\mathsf{sk}_2, m$)**

- Assume a power-of-2 cyclotomic ring
  $R = \mathbb{Z}[X]/(X^N + 1)$
- First round can be computed offline!
- $b^{(j)}$ follows Gaussian $D_{\sigma_b}$
- Hard to predict $\mathbf{u}$ without querying the RO
- Signature size in the $n$-party case:
  $O(\log(N \cdot n))$ larger than a single-user FSwA

$m, (\mathbf{u}, \mathbf{z})$ →

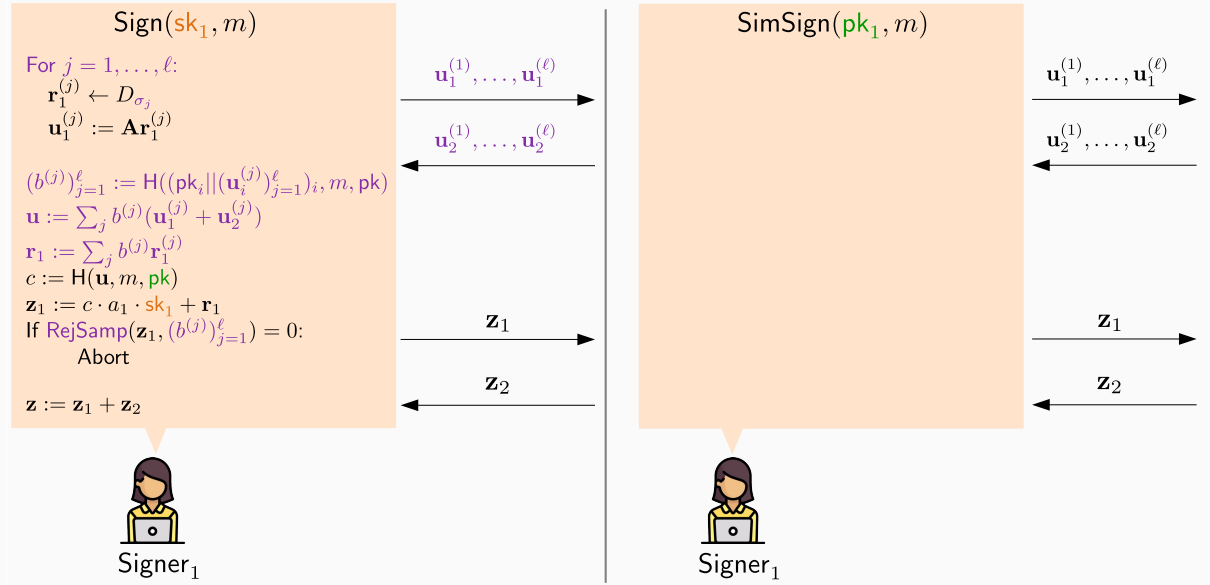
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
Accept iff
$\mathbf{A}\mathbf{z} = c \cdot \mathsf{pk} + \mathbf{u}$
$\wedge \|\mathbf{z}\| \leq \sqrt{2} \cdot B'$

Signer$_1$

Signer$_2$

Verifier

# Key Techniques to Simulate Honest Signer



# Key Techniques to Simulate Honest Signer

## $\mathsf{Sign}(\mathsf{sk}_1, m)$

For $j = 1, \ldots, \ell$:
$\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
$\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$
If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathsf{Signer}_1$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

## $\mathsf{SimSign}(\mathsf{pk}_1, m)$

1. Generalized rejection sampling
   - Fixes $\sigma$ of $\mathbf{z}_1$

$\mathsf{Signer}_1$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

# Key Techniques to Simulate Honest Signer

## $\mathsf{Sign}(\mathsf{sk}_1, m)$

For $j = 1, \ldots, \ell$:
- $\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
- $\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$

$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$

$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$

$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$

$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$

If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

$\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$ ←

$\mathbf{z}_1$ →

$\mathbf{z}_2$ ←

$\mathsf{Signer}_1$

## $\mathsf{SimSign}(\mathsf{pk}_1, m)$

1. Generalized rejection sampling
   - Fixes $\sigma$ of $\mathbf{z}_1$
2. Preimage sampling
   - Generate $\mathbf{u}_1^{(j)}$ with a known trapdoor

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

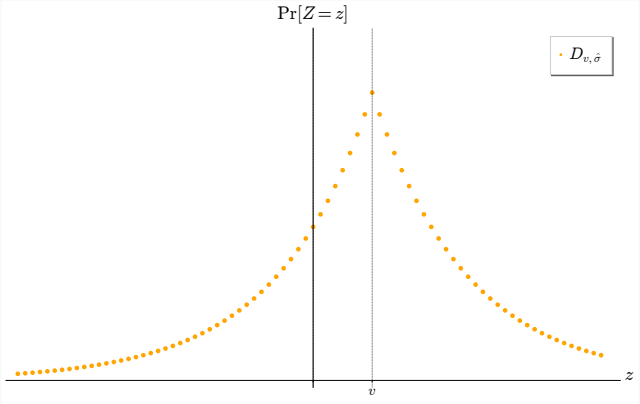
$\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$ ←

$\mathbf{z}_1$ →

$\mathbf{z}_2$ ←

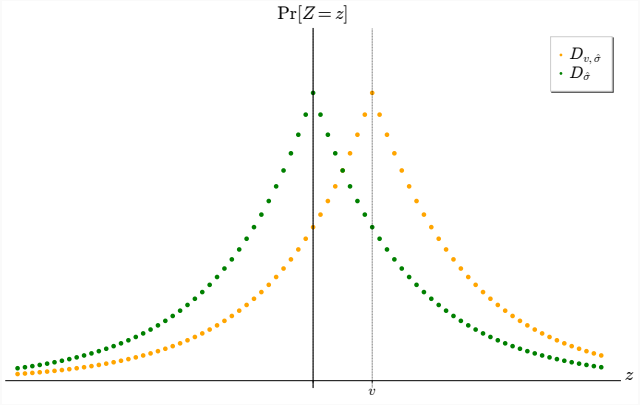
$\mathsf{Signer}_1$

# Standard Rejection Sampling [Lyu12]



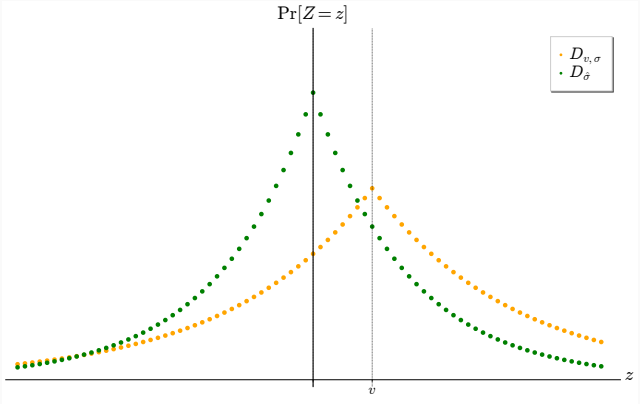

$\mathbf{v} := c \cdot \mathsf{sk}$
$\mathbf{z} \leftarrow D_{\mathbf{v},\hat{\sigma}}$
With prob. $\min\{D_{\hat{\sigma}}(\mathbf{z})/(M \cdot D_{\mathbf{v},\hat{\sigma}}(\mathbf{z})), 1\}$
**return** $\mathbf{z}$

# Standard Rejection Sampling [Lyu12]




$\mathbf{v} := c \cdot \mathsf{sk}$
$\mathbf{z} \leftarrow D_{\mathbf{v},\hat{\sigma}}$
With prob. $\min\{D_{\hat{\sigma}}(\mathbf{z})/(M \cdot D_{\mathbf{v},\hat{\sigma}}(\mathbf{z})), 1\}$
**return** $\mathbf{z}$

# Key Technique I: Generalized Rejection Sampling

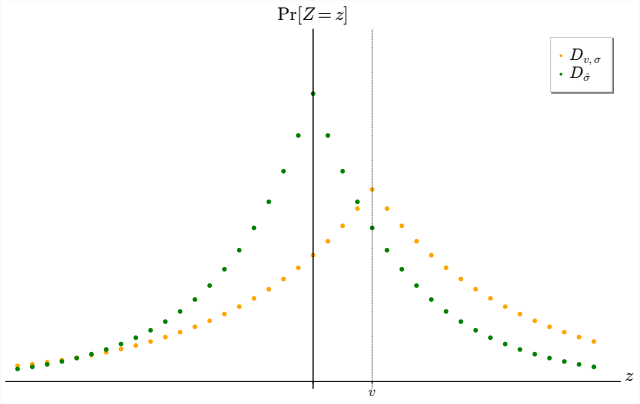



$\mathbf{v} := c \cdot \mathsf{sk}$
$\mathbf{z} \leftarrow D_{\mathbf{v},\sigma}$
With prob. $\min\{D_{\hat{\sigma}}(\mathbf{z})/(M \cdot D_{\mathbf{v},\sigma}(\mathbf{z})), 1\}$
**return** $\mathbf{z}$

In MuSig-L:

- $\sigma$ depends on random coefficients $b^{(j)}$
- Output $\mathbf{z} \sim D_{\hat{\sigma}}$ must be independent of $b^{(j)}$'s

# Key Technique I: Generalized Rejection Sampling

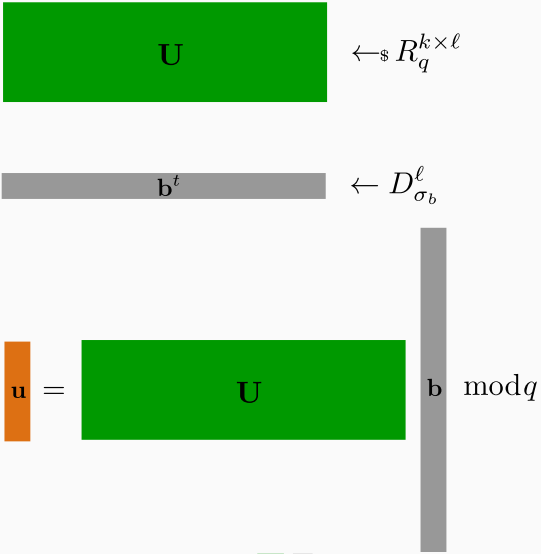



$\mathbf{v} := c \cdot \mathsf{sk}$
$\mathbf{z} \leftarrow D_{\mathbf{v},\sigma,\Lambda+\mathbf{u}}$
With prob. $\min\{D_{\hat{\sigma}}(\mathbf{z})/(M \cdot D_{\mathbf{v},\sigma}(\mathbf{z})), 1\}$
**return** $\mathbf{z}$

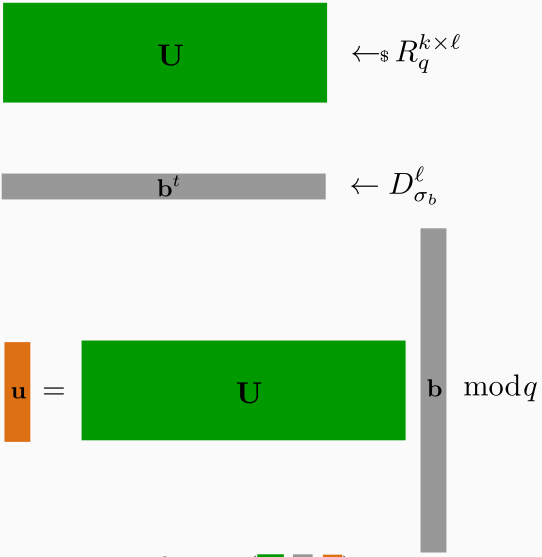
In MuSig-L:

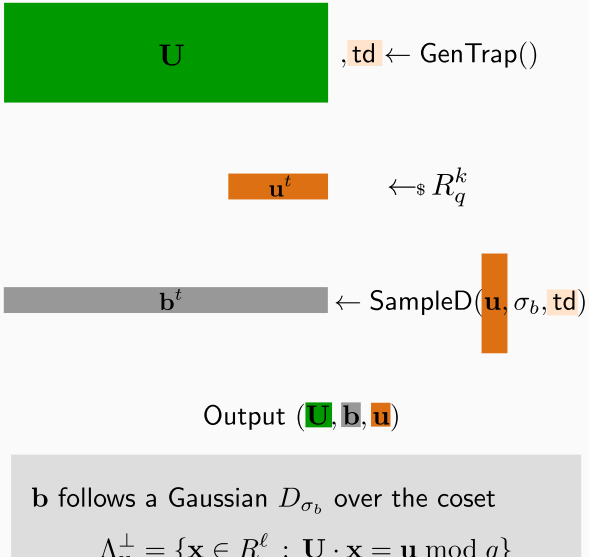
- $\sigma$ depends on random coefficients $b^{(j)}$
- Output $\mathbf{z} \sim D_{\hat{\sigma}}$ must be independent of $b^{(j)}$'s

# Key Technique II: Preimage Sampling with a Lattice Trapdoor [Ajt99,AP09,GPV08,MP12,...]



# Key Technique II: Preimage Sampling with a Lattice Trapdoor [Ajt99,AP09,GPV08,MP12,...]




$\mathbf{b}$ follows a Gaussian $D_{\sigma_b}$ over the coset

$$\Lambda_{\mathbf{u}}^{\perp} = \{\mathbf{x} \in R^{\ell} \,:\, \mathbf{U}\cdot\mathbf{x} = \mathbf{u} \bmod q\}$$

# Putting Them Together: Sign Oracle Simulation

## $\mathsf{Sign}(\mathsf{sk}_1, m)$

For $j = 1, \ldots, \ell$:
$\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
$\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$
If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →
← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$
$\mathbf{z}_1$ →
← $\mathbf{z}_2$

$\mathsf{Signer}_1$

## $\mathsf{SimSign}(\mathsf{pk}_1, m)$

$(\mathbf{U}, \mathsf{td}) \leftarrow \mathsf{GenTrap}()$
$(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}) := \mathbf{U}$

$(\mathbf{u}_1, c, \mathbf{z}_1) \leftarrow \mathsf{ZKSim}(\mathsf{pk}_1)$
$(b^{(j)})_{j=1}^{\ell} := \mathbf{b} \leftarrow \mathsf{SampleD}(\mathbf{u}_1, \sigma_b, \mathsf{td})$
$\mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk}) := \mathbf{b}$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathsf{H}(\mathbf{u}, m, \mathsf{pk}) := c$
With prob. $1/M$:
Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →
← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$
$\mathbf{z}_1$ →
← $\mathbf{z}_2$

$\mathsf{Signer}_1$

- SimSign generates $\mathbf{u}_1^{(j)}$ with trapdoors
- ZKSim ensures $\mathbf{A} \cdot \mathbf{z}_1 = c \cdot a_1 \cdot \mathsf{pk}_1 + \mathbf{u}_1$
- Given td, sample $\mathbf{b}$ s.t. $\mathbf{U} \cdot \mathbf{b} = \mathbf{u}_1$
- **[MP12]** trapdoor: $\ell \approx \log q$

# Putting Them Together: Sign Oracle Simulation

## $\mathsf{Sign}(\mathsf{sk}_1, m)$

For $j = 1, \ldots, \ell$:
$\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
$\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

$\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$ ←

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$
If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{z}_1$ →

$\mathbf{z}_2$ ←

$\mathsf{Signer}_1$

## $\mathsf{SimSign}(\mathsf{pk}_1, m)$

$(\mathbf{U}, \mathsf{td}) \leftarrow \mathsf{GenTrap}()$
$(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}) := \mathbf{U}$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

$\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$ ←

$(\mathbf{u}_1, c, \mathbf{z}_1) \leftarrow \mathsf{ZKSim}(\mathsf{pk}_1)$
$(b^{(j)})_{j=1}^{\ell} := \mathbf{b} \leftarrow \mathsf{SampleD}(\mathbf{u}_1, \sigma_b, \mathsf{td})$
$\mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk}) := \mathbf{b}$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathsf{H}(\mathbf{u}, m, \mathsf{pk}) := c$
With prob. $1/M$:
Abort

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

$\mathbf{z}_1$ →

$\mathbf{z}_2$ ←

$\mathsf{Signer}_1$

- SimSign generates $\mathbf{u}_1^{(j)}$ with trapdoors
- ZKSim ensures $\mathbf{A} \cdot \mathbf{z}_1 = c \cdot a_1 \cdot \mathsf{pk}_1 + \mathbf{u}_1$
- Given td, sample $\mathbf{b}$ s.t. $\mathbf{U} \cdot \mathbf{b} = \mathbf{u}_1$
- **[MP12]** trapdoor: $\ell \approx \log q$

# Putting Them Together: Sign Oracle Simulation

## $\mathsf{Sign}(\mathsf{sk}_1, m)$

For $j = 1, \ldots, \ell$:
$\mathbf{r}_1^{(j)} \leftarrow D_{\sigma_j}$
$\mathbf{u}_1^{(j)} := \mathbf{A}\mathbf{r}_1^{(j)}$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$

$(b^{(j)})_{j=1}^{\ell} := \mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk})$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathbf{r}_1 := \sum_j b^{(j)}\mathbf{r}_1^{(j)}$
$c := \mathsf{H}(\mathbf{u}, m, \mathsf{pk})$
$\mathbf{z}_1 := c \cdot a_1 \cdot \mathsf{sk}_1 + \mathbf{r}_1$
If $\mathsf{RejSamp}(\mathbf{z}_1, (b^{(j)})_{j=1}^{\ell}) = 0$:
Abort

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

Signer$_1$

## $\mathsf{SimSign}(\mathsf{pk}_1, m)$

$(\mathbf{U}, \mathsf{td}) \leftarrow \mathsf{GenTrap}()$
$(\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}) := \mathbf{U}$

$\mathbf{u}_1^{(1)}, \ldots, \mathbf{u}_1^{(\ell)}$ →

← $\mathbf{u}_2^{(1)}, \ldots, \mathbf{u}_2^{(\ell)}$

$(\mathbf{u}_1, c, \mathbf{z}_1) \leftarrow \mathsf{ZKSim}(\mathsf{pk}_1)$
$(b^{(j)})_{j=1}^{\ell} := \mathbf{b} \leftarrow \mathsf{SampleD}(\mathbf{u}_1, \sigma_b, \mathsf{td})$
$\mathsf{H}((\mathsf{pk}_i || (\mathbf{u}_i^{(j)})_{j=1}^{\ell})_i, m, \mathsf{pk}) := \mathbf{b}$
$\mathbf{u} := \sum_j b^{(j)}(\mathbf{u}_1^{(j)} + \mathbf{u}_2^{(j)})$
$\mathsf{H}(\mathbf{u}, m, \mathsf{pk}) := c$
With prob. $1/M$:
Abort

$\mathbf{z}_1$ →

← $\mathbf{z}_2$

$\mathbf{z} := \mathbf{z}_1 + \mathbf{z}_2$

Signer$_1$

- SimSign generates $\mathbf{u}_1^{(j)}$ with trapdoors
- ZKSim ensures $\mathbf{A} \cdot \mathbf{z}_1 = c \cdot a_1 \cdot \mathsf{pk}_1 + \mathbf{u}_1$
- Given td, sample $\mathbf{b}$ s.t. $\mathbf{U} \cdot \mathbf{b} = \mathbf{u}_1$
- **[MP12]** trapdoor: $\ell \approx \log q$

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: https://ia.cr/2022/1036

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: https://ia.cr/2022/1036

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: https://ia.cr/2022/1036

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: https://ia.cr/2022/1036

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: https://ia.cr/2022/1036

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no “one-more” assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: **https://ia.cr/2022/1036**

# Takeaways

- Feasibility of **FSwA multi-signature with single-round online phase**
  - Statistical simulation of sign oracle (no "one-more" assumption!)
  - Forking lemma to show a reduction to M-SIS and M-LWE in the classical ROM
- Key observations:
  1. Generalized rejection sampling lemma
  2. Preimage sampling using a lattice trapdoor (only in the security proof)

## Concurrent Work & Open Questions

- **Squirrel** [FSZ22]: Synchronized MS from OTS + Merkle tree
- Efficient instantiation: exploit **NTRU** or **one-more SIS** [AKSY21] to minimize the overhead in signature size & communication?
- Proof in the QROM & simulation-based security

**Thank you!** – ePrint: **https://ia.cr/2022/1036**

Freepik.
Icons made by Freepik from Flaticon.com.
http://www.flaticon.com.