

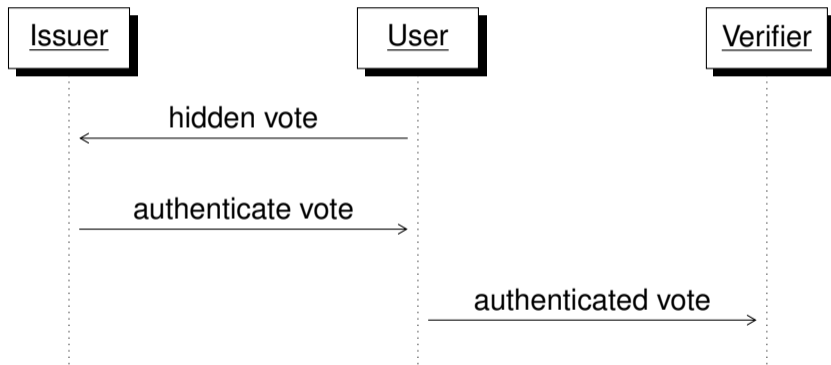
**DTU**



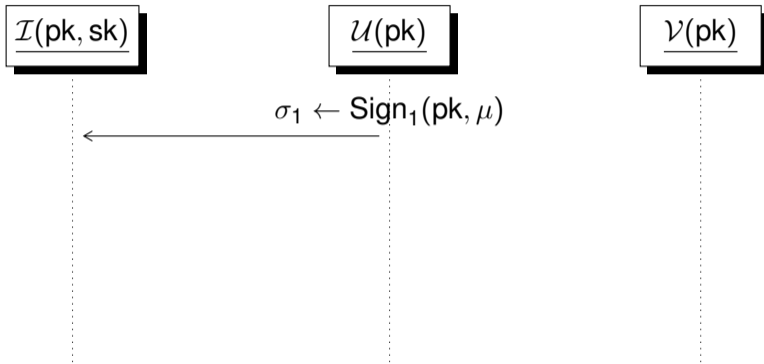
Carsten Baum, **Marvin Beckmann**, Ward Beullens, Shibam Mukherjee, Christian Rechberger

# Blind Signatures Based on **VOLEith** and **MAYO**

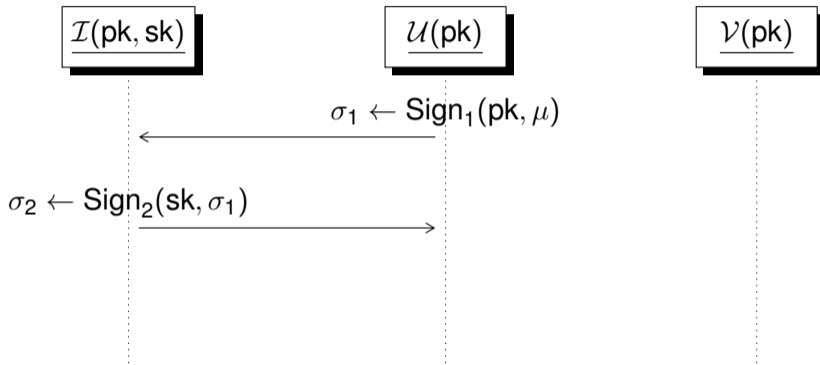
## Example: E-Voting with Blind Signatures



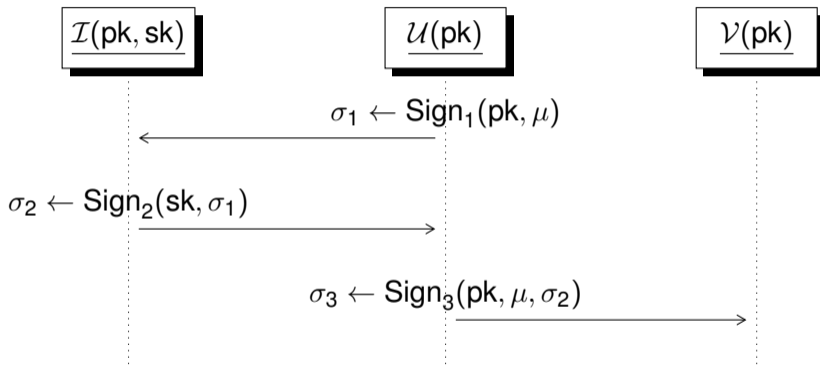
# Blind Signatures



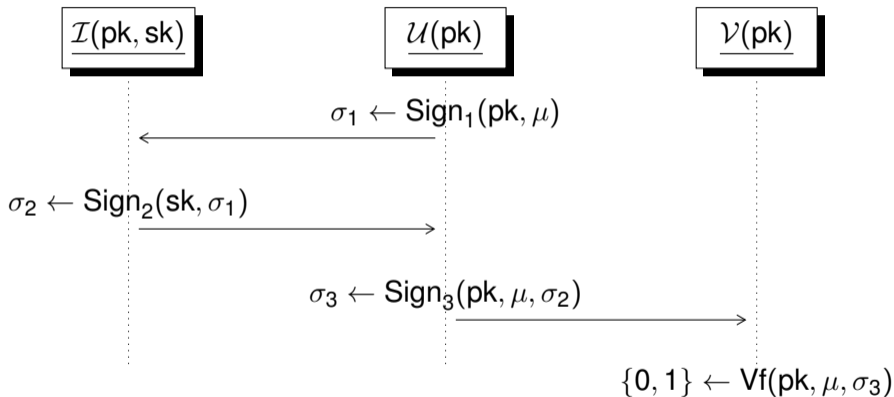
# Blind Signatures



# Blind Signatures

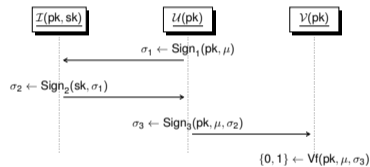


# Blind Signatures



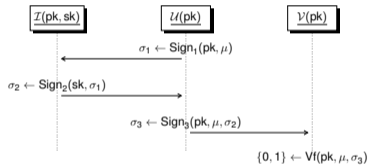
# Blindness Property of Blind Signatures

- $\sigma_1$  and  $\sigma_3$  can not be associated with each other



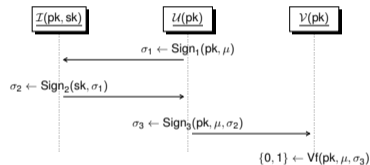
# Fischlin's Blueprint for Blind Signatures

- $(pk, sk)$  are from a normal signature scheme



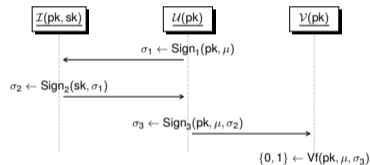
# Fischlin's Blueprint for Blind Signatures

- $(pk, sk)$  are from a normal signature scheme
- $\sigma_1 \leftarrow Com(\mu, r)$  where  $r \leftarrow \{0, 1\}^{1\lambda}$



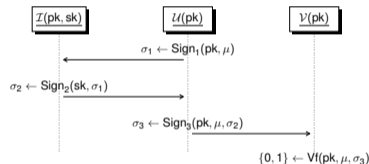
# Fischlin's Blueprint for Blind Signatures

- $(pk, sk)$  are from a normal signature scheme
- $\sigma_1 \leftarrow Com(\mu, r)$  where  $r \leftarrow \{0, 1\}^{1\lambda}$
- $\sigma_2 \leftarrow Sign(sk, \sigma_1)$



# Fischlin's Blueprint for Blind Signatures

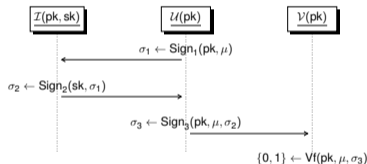
- $(pk, sk)$  are from a normal signature scheme
- $\sigma_1 \leftarrow Com(\mu, r)$  where  $r \leftarrow \{0, 1\}^{1\lambda}$
- $\sigma_2 \leftarrow Sign(sk, \sigma_1)$
- $\sigma_3 \leftarrow (\mu, z)$  where  $z \leftarrow NIZK(pk, \sigma_2, \mu, r)$



# High-Level Description of our Blind Signature

For a fixed  $\mu$ , and fixed MAYO keys  $(pk, sk)$

- 1 :  $r \leftarrow_{\$} \{0, 1\}^\lambda$
- 2 :  $c \leftarrow \text{SHAKE256}(r \parallel \mu)$  / commitment
- 3 :  $\text{salt} \leftarrow_{\$} \{0, 1\}^\lambda$
- 4 :  $t \leftarrow \text{SHAKE256}(\text{salt} \parallel c)$
- 5 :  $\sigma \leftarrow \text{MAYO.SamplePre}(sk, pk, t)$  / signing
- 6 :  $\text{VOLEitH-ZKP}(\mu, r, \text{salt}, \sigma)$  / showing



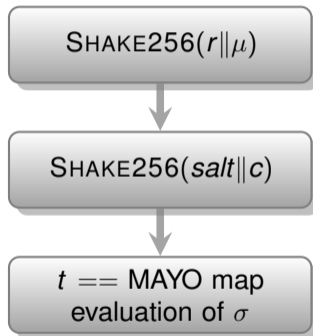
# **VOLeith Proof System and How our Proof is Constructed**

## Properties of VOLEith as a Proof System

- Commit to witness before generating the proof.
- Operations:
  - Additions are linear and performed locally without additional cost.
  - Multiplications require non-linear consistency checks; creates a tradeoff between prover computation time and communication overhead.
- Consistency-Check:
  - Allows proving that different commitments refer to the same underlying secret

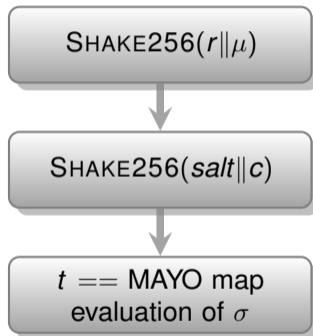
## ZKP Strategy

- 1 bit-commit to  $r||\mu$
- 2 SHAKE256 with bit-commit and VOLEitH properties
- 3 SHAKE256 with previous result and bit-commit to salt
- 4  $\mathbb{F}_{16}$  commitments of  $\sigma$
- 5 MAYO map evaluation
- 6 Compare with consistency-check

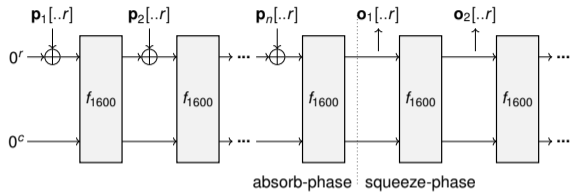


# ZKP Strategy

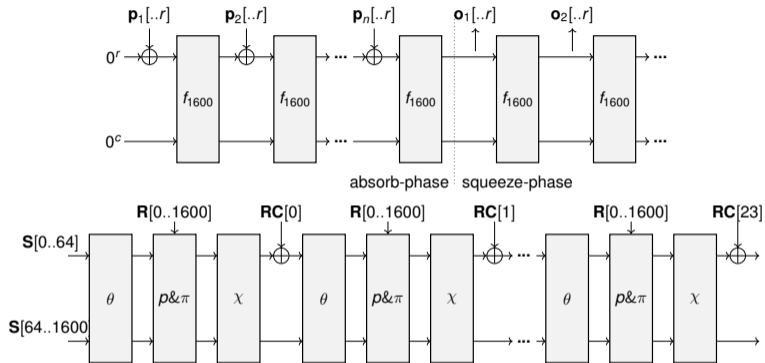
- 1 bit-commit to  $r||\mu$
- 2 SHAKE256 with bit-commit and VOLEitH properties
- 3 SHAKE256 with previous result and bit-commit to salt
- 4  $\mathbb{F}_{16}$  commitments of  $\sigma$
- 5 MAYO map evaluation
- 6 Compare with consistency-check



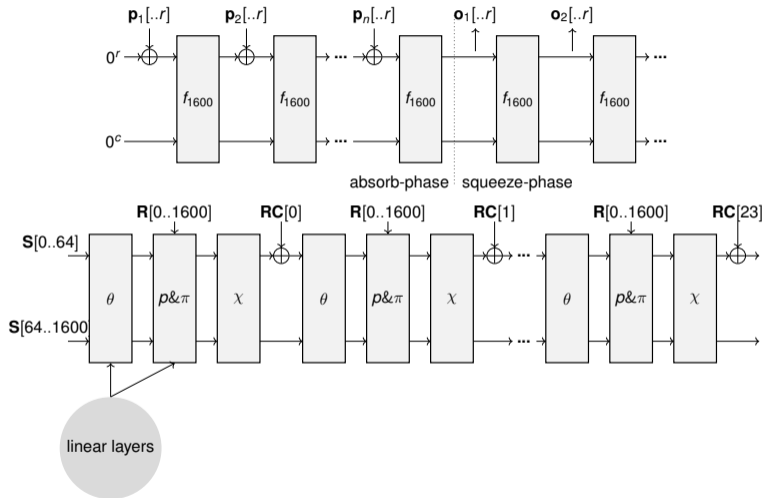
# SHAKE256



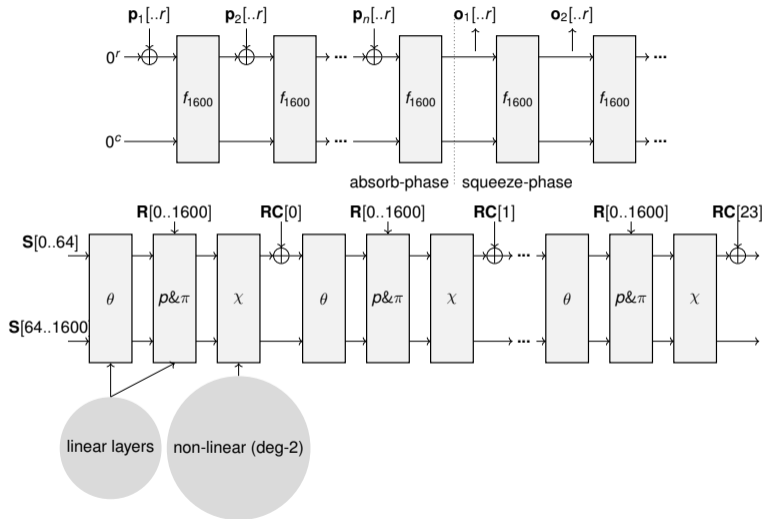
# SHAKE256



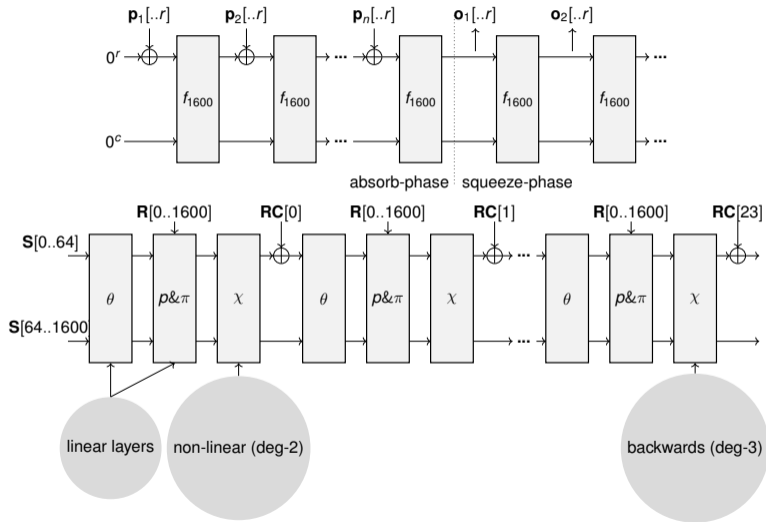
# SHAKE256



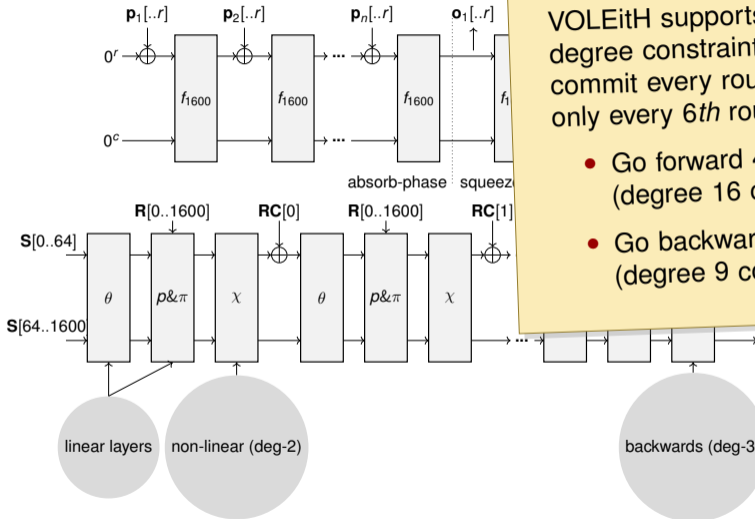
# SHAKE256



# SHAKE256



# SHAKE256



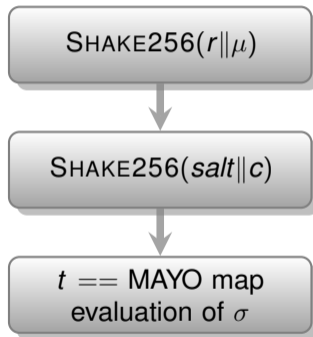
## Optimization Strategy:

VOLEitH supports higher-degree constraints: Do not commit every round, but only every 6th round.

- Go forward 4 steps (degree 16 constraints)
- Go backward 2 steps (degree 9 constraints)

## ZKP Strategy

- 1 bit-commit to  $r||\mu$
- 2 SHAKE256 with bit-commit and VOLEitH properties
- 3 SHAKE256 with previous result and bit-commit to salt
- 4  $\mathbb{F}_{16}$  commitments of  $\sigma$
- 5 MAYO map evaluation
- 6 Compare with consistency-check



## MAYO Map Evaluation

- There are  $n$  public matrices  $\mathbf{M}_1, \dots, \mathbf{M}_n \in \mathbb{F}_q^{m \times m}$
- The signature  $\vec{s}$  is generated such that

$$\forall i \in [n] : v_i = \vec{s}^\top \mathbf{M}_i \vec{s}$$

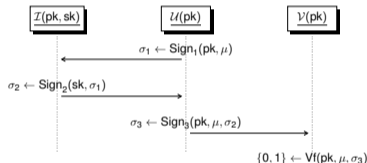
where  $H(\mu) = \vec{v} \in \mathbb{F}_q^n$

▷ Verifying the signature is only a degree-2 operation on the signature  $\vec{s}$

# VOLEitH as a Proof System for Fischlin-Style ZKP

For a fixed  $\mu$ , and fixed MAYO keys  $(pk, sk)$

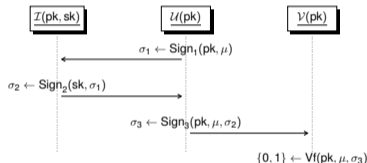
- 1 :  $r \leftarrow_{\$} \{0, 1\}^\lambda$
- 2 :  $c \leftarrow \text{SHAKE256}(r \parallel \mu)$  / commitment
- 3 :  $salt \leftarrow_{\$} \{0, 1\}^\lambda$
- 4 :  $t \leftarrow \text{SHAKE256}(salt \parallel c)$
- 5 :  $\sigma \leftarrow \text{MAYO.SamplePre}(sk, pk, t)$  / signing
- 6 :  $\text{VOLEitH-ZKP}(\mu, r, salt, \sigma)$  / showing



# VOLEitH as a Proof System for Fischlin-Style ZKP

For a fixed  $\mu$ , and fixed MAYO keys  $(pk, sk)$

- 1 :  $r \leftarrow_{\$} \{0, 1\}^\lambda$
- 2 :  $c \leftarrow \text{SHAKE256}(r \parallel \mu)$  / commitment
- 3 :  $salt \leftarrow_{\$} \{0, 1\}^\lambda$
- 4 :  $t \leftarrow \text{SHAKE256}(salt \parallel c)$
- 5 :  $\sigma \leftarrow \text{MAYO.SamplePre}(sk, pk, t)$  / signing
- 6 :  $\text{VOLEitH-ZKP}(\mu, r, salt, \sigma)$  / showing



Slightly unsatisfied: SHAKE256 takes up a large portion of runtime and size

## Avoiding SHAKE256

- 1 Replace SHAKE256 with RAINHASH

# Avoiding SHAKE256

- 1 Replace SHAKE256 with RAINHASH
  - 7 rounds of the RAIN permutation
  - each round consists of field inversion (non-linear) + matrix vector multiplication with a fixed matrix (linear)
  - we commit after every round and make consistency-checks for the inverse
  - only 7 additional commitments of the state + very efficient

## Avoiding SHAKE256

- 1 Replace SHAKE256 with RAINHASH
  - 7 rounds of the RAIN permutation
  - each round consists of field inversion (non-linear) + matrix vector multiplication with a fixed matrix (linear)
  - we commit after every round and make consistency-checks for the inverse
  - only 7 additional commitments of the state + very efficient
- 2 Remove need to prove hashes in ZKP using stronger assumptions on MAYO + useful properties of VOLEith

# Competitiveness

Scheme	Assumptions	Signing (in KB)	Showing (in KB)	Runtime Showing (in ms)	Security (in bits)
Agrawal et al. [AKSY22]	One-More-ISIS	1.4	45	–	109
del Pino <sup>a</sup> & Katsumata [dPK22]	MSIS/MLWE/NTRU	851	102.6	–	128
Lyubashevsky <sup>b</sup> et al. [LNP22]	MSIS/MLWE	$\geq 1000$	150	–	128
Bootle et al. [BLNS23b]	ISIS-f/NTRU	$> 100$	$> 100$	–	128
Policharla et al. [PFW23]	MSIS/MLWE+Poseidon	2.4	85 – 173	304 – 4822	115
Lyubashevsky <sup>c</sup> et al. [LSS24]	ISIS-f/NTRU	1.3	29	$< 200$	128
Argo et al. [AGJ <sup>+</sup> 24] (implementing [JRS23])	MSIS/MLWE	45	79.6	$< 500$	128
Beullens <sup>d</sup> et al. [BLNS23a]	MSIS/MLWE/NTRU	60	22	–	128
Jeudy and Sanders [JS24]	MSIS/MLWE	60	41	–	128
Baldimtsi <sup>e</sup> et al. [BCGY24]	One-More-ISIS	1	68	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25a] <sup>f</sup>	WAVE	–	39-138 <sup>f</sup>	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25a] <sup>f</sup>	UOV	–	3.8-11 <sup>f</sup>	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25b]	One-More Quadratic Claw	–	6.8	–	128
SHAKE256-deg16+MAYO-128 <sub>s/f</sub>	MAYO	0.486	24.3/41.6	178.1/75.6	128
RainHash+MAYO-128 <sub>s/f</sub>	MAYO	0.486	16.5/27.8	114.4/49.8	128
One-More-MAYO-128 <sub>s/f</sub>	One-More-MAYO	0.469	6.7/10.4	42/40	128

<sup>a</sup> May be improved upon using more recent lattice NIZKs.

<sup>b</sup> Bounded number of signatures, 1.3MB public key.

<sup>c</sup> Implements a version of [BLNS23b].

<sup>d</sup> Uses a generic NIZK scheme.

<sup>e</sup> Non-interactive construction with pseudorandom messages.

<sup>f</sup> MB cost in communication between user and issuer.

# Competitiveness

Scheme	Assumptions	Signing (in KB)	Showing (in KB)	Runtime Showing (in ms)	Security (in bits)
Agrawal et al. [AKSY22]	One-More-ISIS	1.4	45	–	109
del Pino <sup>a</sup> & Katsumata [dPK22]	MSIS/MLWE/NTRU	851	102.6	–	128
Lyubashevsky <sup>b</sup> et al. [LNP22]	MSIS/MLWE	$\geq 1000$	150	–	128
Bootle et al. [BLNS23b]	ISIS-f/NTRU	$> 100$	$> 100$	–	128
Policharla et al. [PFW23]	MSIS/MLWE+Poseidon	2.4	85 – 173	304 – 4822	115
Lyubashevsky <sup>c</sup> et al. [LSS24]	ISIS-f/NTRU	1.3	29	$< 200$	128
Argo et al. [AGJ <sup>+</sup> 24] (implementing [JRS23])	MSIS/MLWE	45	79.6	$< 500$	128
Beullens <sup>d</sup> et al. [BLNS23a]	MSIS/MLWE/NTRU	60	22	–	128
Jeudy and Sanders [JS24]	MSIS/MLWE	60	41	–	128
Baldimtsi <sup>e</sup> et al. [BCGY24]	One-More-ISIS	1	68	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25a] <sup>f</sup>	WAVE	–	39-138 <sup>f</sup>	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25a] <sup>f</sup>	UOV	–	3.8-11 <sup>f</sup>	–	128
Bouillaguet et al. [BFM <sup>+</sup> 25b]	One-More Quadratic Claw	–	6.8	–	128
SHAKE256-deg16+MAYO-128 <sub>s/f</sub>	MAYO	0.486	24.3/41.6	178.1/75.6	128
RainHash+MAYO-128 <sub>s/f</sub>	MAYO	0.486	16.5/27.8	114.4/49.8	128
One-More-MAYO-128 <sub>s/f</sub>	One-More-MAYO	0.469	6.7/10.4	42/40	128




<sup>a</sup> May be improved upon using more recent lattice NIZKs. <sup>b</sup> Bounded number of signatures, 1.3MB public key. <sup>c</sup> Implements a version of [BLNS23b].

<sup>d</sup> Uses a generic NIZK scheme. <sup>e</sup> Non-interactive construction with pseudorandom messages. <sup>f</sup> MB cost in communication between user and issuer.




Thanks for listening! Paper on eprint: 2026/109



## References I

-  Sven Argo, Tim Güneysu, Corentin Jeudy, Georg Land, Adeline Roux-Langlois, and Olivier Sanders.  
Practical post-quantum signatures for privacy.  
pages 1523–1537, 2024.
-  Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav.  
Practical, round-optimal lattice-based blind signatures.  
pages 39–53, 2022.
-  Foteini Baldimtsi, Jiaqi Cheng, Rishab Goyal, and Aayush Yadav.  
Non-interactive blind signatures: Post-quantum and stronger security.  
pages 70–104, 2024.

## References II

-  Charles Bouillaguet, Thibault Feneuil, Jules Maire, Matthieu Rivain, Julia Sauvage, and Damien Vergnaud.  
Blinding post-quantum hash-and-sign signatures.  
[Cryptology ePrint Archive, Paper 2025/895, 2025.](#)
-  Charles Bouillaguet, Thibault Feneuil, Jules Maire, Matthieu Rivain, Julia Sauvage, and Damien Vergnaud.  
Multivariate commitments and signatures with efficient protocols.  
[Cryptology ePrint Archive, Paper 2025/2035, 2025.](#)
-  Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler.  
Lattice-based blind signatures: Short, efficient, and round-optimal.  
[pages 16–29, 2023.](#)

## References III



Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti.

A framework for practical anonymous credentials from lattices.  
pages 384–417, 2023.



Rafaël del Pino and Shuichi Katsumata.

A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling.  
pages 306–336, 2022.



Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders.

Lattice signature with efficient protocols, application to anonymous credentials.  
pages 351–383, 2023.

## References IV



Corentin Jeudy and Olivier Sanders.

Improved lattice blind signatures from recycled entropy.  
[Cryptology ePrint Archive, Report 2024/1289, 2024.](#)



Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon.

Efficient lattice-based blind signatures via gaussian one-time signatures.  
[pages 498–527, 2022.](#)



Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer.

The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy.  
[pages 3125–3137, 2024.](#)

## References V



Guru-Vamsi Policharla, Bas Westerbaan, Armando Faz-Hernández, and Christopher A. Wood.

Post-quantum privacy pass via post-quantum anonymous credentials.  
*IACR Cryptol. ePrint Arch.*, page 414, 2023.

**Extra Slides**

## Removing The Hash Calls

- Achieve the hiding not through a commitment scheme, but inherent randomness from the proof generation:

$$t = H(\mu, \pi_1) \oplus r$$

where  $(\pi_1, r) \leftarrow \mathcal{P}_1$ . This only adds small overhead to the proof itself.

- $\pi_1$  itself fixes  $r$  (it is part of one of the VOLE correlations), so it is fixed before the target is determined and correlation of  $\pi_1$  that fixes  $r$  can then be used to add the bitmask inside of the verification
  - Use a new assumption: One-More-MAYO, that allows querying the MAYO map for chosen targets
- ▷ Less conservative option, but it significantly improves runtime and size. (Primarily impacts for smaller security parameters)