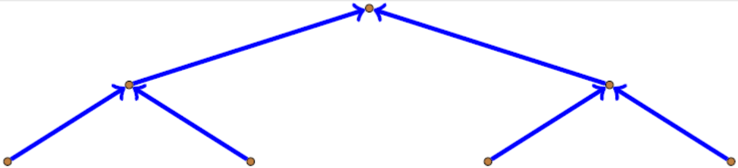
Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures without Trapdoors



Benoît Libert[1] San Ling[2] **Khoa Nguyen**[2] Huaxiong Wang[2]

[1]Ecole Normale Supérieure de Lyon (France)

[2]Nanyang Technological University (Singapore)

EUROCRYPT 2016 - Vienna, Austria

Outline

1. Introduction
2. Our Accumulator and Its Supporting Zero-Knowledge Argument
3. Applications to Ring and Group Signatures

Cryptographic Accumulators

Accumulator [BdM'93]: a function hashing a large data set $R = \{d_0, \ldots, d_{N-1}\}$ into a constant-size value u.

- For any $d \in R$, there is a short witness w that d was accumulated into u.
- It is infeasible to compute a valid witness w^* for some $d^* \notin R$.
- Numerous applications in authentication mechanisms.
- In many scenarios, a ZK proof of an input-witness pair (d, w) is desirable.

Previous Works

- 2 main families of number-theoretic accumulators: based on groups of hidden order, or on pairings (strong RSA and strong DH assumptions).
- A 3^{rd} family relies on Merkle trees: hardly compatible with ZK proofs.
  - Known methods require non-standard assumptions in groups of hidden order [BCG'14] or non-falsifiable knowledge assumptions [BSCG+'14].
  - [PSTY'13]: SIS-based Merkle tree; supporting ZK proofs were not considered.

Our Results

First lattice-based accumulator supported by logarithmic-size ZK arguments.

- We build Merkle trees from a family of SIS-based CRHF

$$\mathcal{H} : D \times D \to D.$$

- We demonstrate in ZK the possession of a Merkle tree path (hash chain).

Our Results

First lattice-based accumulator supported by logarithmic-size ZK arguments.

- We build Merkle trees from a family of SIS-based CRHF

$$\mathcal{H} : D \times D \rightarrow D.$$

- We demonstrate in ZK the possession of a Merkle tree path (hash chain).

Applications:

1. First lattice-based logarithmic-size ring signature.

Our Results

First lattice-based accumulator supported by logarithmic-size ZK arguments.

- We build Merkle trees from a family of SIS-based CRHF

$$\mathcal{H} : D \times D \to D.$$

- We demonstrate in ZK the possession of a Merkle tree path (hash chain).

Applications:

1. First lattice-based logarithmic-size ring signature.
2. First group signature without lattice trapdoors. Previous constructions [GKV'10,CNR'12,LLLS'13,LNW'15,NZZ'15] rely on trapdoors for key generation and/or for enabling tracing.

   Being trapdoor-less: smaller parameters, shorter key and signature sizes.

   User's signing key in our scheme has size of several KBs, compared with ≈ 90 GBs in [NZZ'15].

Outline

1. Introduction
2. Our Accumulator and Its Supporting Zero-Knowledge Argument
3. Applications to Ring and Group Signatures

A Family of Lattice-Based CRHF

Let n be the security parameter, $q = \widetilde{\mathcal{O}}(n)$, $k = \lceil \log_2 q \rceil$, and $m = 2nk$. Define:

$$\mathbf{G} = \begin{bmatrix} 1\ 2\ 4\ \ldots\ 2^{k-1} & & \\ & \ldots & \\ & & 1\ 2\ 4\ \ldots\ 2^{k-1} \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}.$$

For all $\mathbf{v} \in \mathbb{Z}_q^n$: $\mathbf{v} = \mathbf{G} \cdot \mathsf{bin}(\mathbf{v})$, where $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{nk}$ - the bin. rep. of $\mathbf{v}$.

A Family of Lattice-Based CRHF

Let n be the security parameter, $q = \widetilde{\mathcal{O}}(n)$, $k = \lceil \log_2 q \rceil$, and $m = 2nk$. Define:

$$\mathbf{G} = \begin{bmatrix} 1\ 2\ 4\ \ldots\ 2^{k-1} & & \\ & \ldots & \\ & & 1\ 2\ 4\ \ldots\ 2^{k-1} \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}.$$

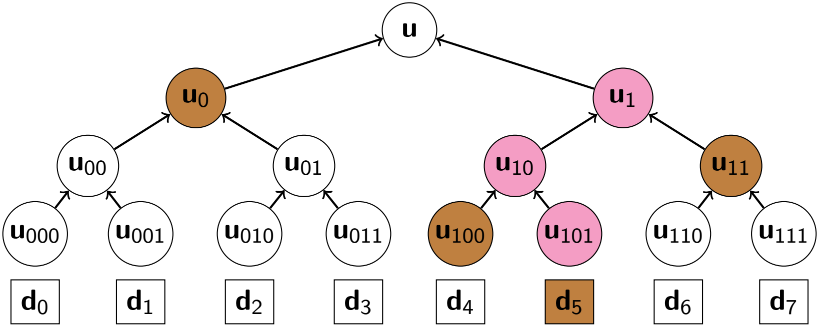
For all $\mathbf{v} \in \mathbb{Z}_q^n$: $\mathbf{v} = \mathbf{G} \cdot \mathsf{bin}(\mathbf{v})$, where $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{nk}$ - the bin. rep. of $\mathbf{v}$.

Define the family $\mathcal{H} : \{0,1\}^{nk} \times \{0,1\}^{nk} \rightarrow \{0,1\}^{nk}$ as $\mathcal{H} = \{h_{\mathbf{A}} \mid \mathbf{A} \in \mathbb{Z}_q^{n \times m}\}$, where for $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1]$ with $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times nk}$, and $(\mathbf{u}_0, \mathbf{u}_1) \in \{0,1\}^{nk} \times \{0,1\}^{nk}$,

$$h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathsf{bin}\big(\mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 \bmod q\big) \in \{0,1\}^{nk}.$$

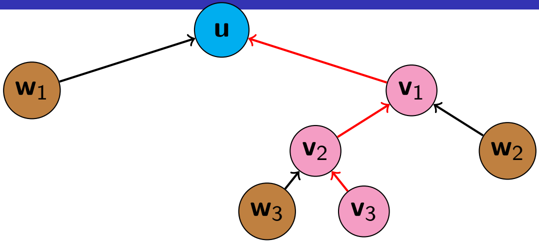
- Note that $h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{u} \Leftrightarrow \mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 = \mathbf{G} \cdot \mathbf{u} \bmod q$.
- $\mathcal{H}$ is collision-resistant, assuming that $\mathsf{SIS}^{\infty}_{n,m,q,1}$ is hard.

From CRHF to Merkle-tree-style Accumulators



- A Merkle tree with $2^3 = 8$ leaves, which accumulates the data blocks $\mathbf{d}_0, \ldots, \mathbf{d}_7$ into the value $\mathbf{u}$ at the root.
- The value at each non-leaf node is the hash of its two children.
- The brown nodes together with the bit string $(j_3, j_2, j_1) = (1, 0, 1)$ form a witness to the fact that $\mathbf{d}_5$ is accumulated into $\mathbf{u}$.

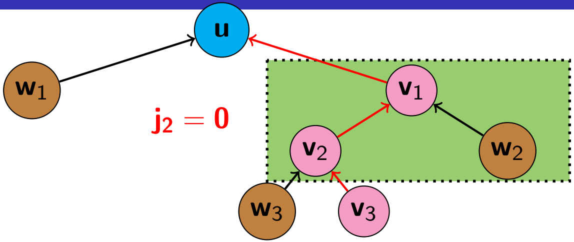
Proving Knowledge of an Accumulated Value



- **Public input:** $\mathbf{A}; \mathbf{u} = \mathbf{v}_0$.
  Secret input: $(\mathbf{w}_\ell, \ldots, \mathbf{w}_1), (\mathbf{v}_\ell, \ldots, \mathbf{v}_1), (j_\ell, \ldots, j_1)$.
- **Prover's goal:** Proving that

$$\forall\, i \in \{\ell - 1, \ldots, 1, 0\} : \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

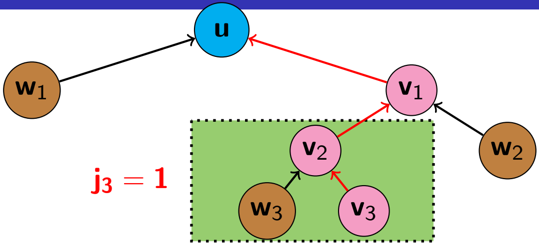
Proving Knowledge of an Accumulated Value



- **Public input:** $\mathbf{A}; \mathbf{u} = \mathbf{v}_0$.
  Secret input: $(\mathbf{w}_\ell, \ldots, \mathbf{w}_1), (\mathbf{v}_\ell, \ldots, \mathbf{v}_1), (j_\ell, \ldots, j_1)$.
- **Prover's goal:** Proving that

$$\forall\, i \in \{\ell-1, \ldots, 1, 0\} :\ \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

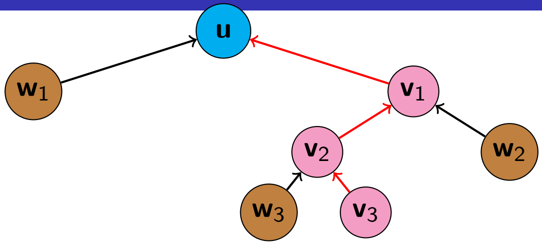
Proving Knowledge of an Accumulated Value



- **Public input:** $\mathbf{A}$; $\mathbf{u} = \mathbf{v}_0$.
  Secret input: $(\mathbf{w}_\ell, \ldots, \mathbf{w}_1), (\mathbf{v}_\ell, \ldots, \mathbf{v}_1), (j_\ell, \ldots, j_1)$.
- **Prover's goal:** Proving that

$$\forall\, i \in \{\ell - 1, \ldots, 1, 0\} : \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

Proving Knowledge of an Accumulated Value



- **Public input:** $\mathbf{A}; \mathbf{u} = \mathbf{v}_0$.

  Secret input: $(\mathbf{w}_\ell, \ldots, \mathbf{w}_1), (\mathbf{v}_\ell, \ldots, \mathbf{v}_1), (j_\ell, \ldots, j_1)$.

- **Prover's goal:** Proving that

$$\forall\, i \in \{\ell - 1, \ldots, 1, 0\} : \mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

✗ Previous protocols for SIS-based hash functions ([Lyu'08,09,12], [LNSW'13]) only prove knowledge of a hidden preimage for a given image.

? Here, we essentially need to prove knowledge of "ℓ hidden preimage-image pairs nested along a hidden path."

Transformations

For any bit b and binary vector $\mathbf{v}$, define $\bar{b} = 1 - b$ and $\mathrm{ext}(b, \mathbf{v}) = \begin{pmatrix} \bar{b} \cdot \mathbf{v} \\ b \cdot \mathbf{v} \end{pmatrix}$.

Transformations

For any bit b and binary vector $\mathbf{v}$, define $\bar{b} = 1 - b$ and $\mathrm{ext}(b, \mathbf{v}) = \begin{pmatrix} \bar{b} \cdot \mathbf{v} \\ b \cdot \mathbf{v} \end{pmatrix}$.

Observe that

$$\mathbf{v}_i = \begin{cases} h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}), & \text{if } j_{i+1} = 0; \\ h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1}), & \text{if } j_{i+1} = 1. \end{cases}$$

is equivalent to:

$$\mathbf{v}_i = \bar{j}_{i+1} \cdot h_{\mathbf{A}}(\mathbf{v}_{i+1}, \mathbf{w}_{i+1}) + j_{i+1} \cdot h_{\mathbf{A}}(\mathbf{w}_{i+1}, \mathbf{v}_{i+1})$$

$$\Leftrightarrow \quad \bar{j}_{i+1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{v}_{i+1} + \mathbf{A}_1 \cdot \mathbf{w}_{i+1}\big) + j_{i+1} \cdot \big(\mathbf{A}_0 \cdot \mathbf{w}_{i+1} + \mathbf{A}_1 \cdot \mathbf{v}_{i+1}\big) = \mathbf{G} \cdot \mathbf{v}_i \bmod q$$

$$\Leftrightarrow \quad \mathbf{A} \cdot \begin{pmatrix} \bar{j}_{i+1} \cdot \mathbf{v}_{i+1} \\ j_{i+1} \cdot \mathbf{v}_{i+1} \end{pmatrix} + \mathbf{A} \cdot \begin{pmatrix} j_{i+1} \cdot \mathbf{w}_{i+1} \\ \bar{j}_{i+1} \cdot \mathbf{w}_{i+1} \end{pmatrix} = \mathbf{G} \cdot \mathbf{v}_i \bmod q$$

$$\Leftrightarrow \quad \mathbf{A} \cdot \mathrm{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A} \cdot \mathrm{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G} \cdot \mathbf{v}_i \bmod q.$$

Developing Stern's Protocol

Now, the task is to prove in ZK the possession of $\{j_i, \mathbf{v}_i, \mathbf{w}_i\}_{i=1}^{\ell}$ s.t.

$$\forall i \in \{\ell - 1, \ldots, 0\} : \mathbf{A} \cdot \mathrm{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A} \cdot \mathrm{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G} \cdot \mathbf{v}_i \bmod q. \quad (1)$$

Developing Stern's Protocol

Now, the task is to prove in ZK the possession of $\{j_i, \mathbf{v}_i, \mathbf{w}_i\}_{i=1}^{\ell}$ s.t.

$$\forall i \in \{\ell-1, \ldots, 0\} : \mathbf{A} \cdot \mathrm{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A} \cdot \mathrm{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G} \cdot \mathbf{v}_i \bmod q. \quad (1)$$

Stern's protocol [Stern'96]: Main ideas

Proving in ZK the possession of a binary vector $\mathbf{s}$ with fixed Hamming weight t, s.t. $\mathbf{M} \cdot \mathbf{s} = \mathbf{u} \bmod q$, for given $(\mathbf{M}, \mathbf{u})$.

1. Proving the linear equation: show that $\mathbf{M}(\mathbf{s}+\mathbf{r}) = \mathbf{u} + \mathbf{M} \cdot \mathbf{r}$ $[q]$, for random $\mathbf{r}$.
2. Proving the constraint of $\mathbf{s}$: show that $\pi(\mathbf{s})$ has weight t, for random π.

Developing Stern's Protocol

Now, the task is to prove in ZK the possession of $\{j_i, \mathbf{v}_i, \mathbf{w}_i\}_{i=1}^{\ell}$ s.t.

$$\forall i \in \{\ell-1, \ldots, 0\} : \mathbf{A}\cdot\mathrm{ext}(j_{i+1}, \mathbf{v}_{i+1}) + \mathbf{A}\cdot\mathrm{ext}(\bar{j}_{i+1}, \mathbf{w}_{i+1}) = \mathbf{G}\cdot\mathbf{v}_i \bmod q. \quad (1)$$

Stern's protocol [Stern'96]: Main ideas

Proving in ZK the possession of a binary vector $\mathbf{s}$ with fixed Hamming weight t, s.t. $\mathbf{M}\cdot\mathbf{s} = \mathbf{u} \bmod q$, for given $(\mathbf{M}, \mathbf{u})$.

1. Proving the linear equation: show that $\mathbf{M}(\mathbf{s}+\mathbf{r}) = \mathbf{u}+\mathbf{M}\cdot\mathbf{r}\ [q]$, for random $\mathbf{r}$.
2. Proving the constraint of $\mathbf{s}$: show that $\pi(\mathbf{s})$ has weight t, for random π.

✓ The first idea can be generalized to prove all ℓ linear equations in (1) hold.

? We'd like to prove the constraints of

$$\mathbf{v}_i \in \{0,1\}^{nk},\ \ \mathbf{w}_i \in \{0,1\}^{nk},\ \ \mathbf{z}_i = \mathrm{ext}(j_i, \mathbf{v}_i)\ \text{ and }\ \mathbf{y}_i = \mathrm{ext}(\bar{j}_i, \mathbf{w}_i)$$

using random permutations. How?

Extensions and Permutations

Proving in ZK that $\mathbf{v}_i, \mathbf{w}_i \in \{0,1\}^{nk}$

1. Extend to $\mathbf{v}_i^*, \mathbf{w}_i^* \in \mathsf{B}_m^{nk}$, res., where $\mathsf{B}_m^{nk} := \{\mathbf{x} \in \{0,1\}^m : \mathsf{wt}(\mathbf{x}) = nk\}$.
2. Show the verifier that $\pi(\mathbf{v}_i^*), \phi(\mathbf{w}_i^*) \in \mathsf{B}_m^{nk}$, where $\pi, \phi \xleftarrow{\$} \mathcal{S}_m$.

Extensions and Permutations

Proving in ZK that $\mathbf{v}_i, \mathbf{w}_i \in \{0,1\}^{nk}$

1. Extend to $\mathbf{v}_i^*, \mathbf{w}_i^* \in \mathsf{B}_m^{nk}$, res., where $\mathsf{B}_m^{nk} := \{\mathbf{x} \in \{0,1\}^m : \mathsf{wt}(\mathbf{x}) = nk\}$.
2. Show the verifier that $\pi(\mathbf{v}_i^*), \phi(\mathbf{w}_i^*) \in \mathsf{B}_m^{nk}$, where $\pi, \phi \xleftarrow{\$} \mathcal{S}_m$.

Proving in ZK that $\mathbf{z}_i^* = \mathsf{ext}(j_i, \mathbf{v}_i^*)$ and $\mathbf{y}_i^* = \mathsf{ext}(\bar{j}_i, \mathbf{w}_i^*)$

1. For $b \in \{0,1\}$, for $\pi \in \mathcal{S}_m$, we define the permutation $F_{b,\pi}$ that transforms vector $\mathbf{z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \end{pmatrix} \in \mathbb{Z}_q^{2m}$ to vector $F_{b,\pi}(\mathbf{z}) = \begin{pmatrix} \pi(\mathbf{z}_b) \\ \pi(\mathbf{z}_{\bar{b}}) \end{pmatrix}$.

Extensions and Permutations

Proving in ZK that $\mathbf{v}_i, \mathbf{w}_i \in \{0,1\}^{nk}$

1. Extend to $\mathbf{v}_i^*, \mathbf{w}_i^* \in \mathsf{B}_m^{nk}$, res., where $\mathsf{B}_m^{nk} := \{\mathbf{x} \in \{0,1\}^m : \mathrm{wt}(\mathbf{x}) = nk\}$.
2. Show the verifier that $\pi(\mathbf{v}_i^*), \phi(\mathbf{w}_i^*) \in \mathsf{B}_m^{nk}$, where $\pi, \phi \xleftarrow{\$} \mathcal{S}_m$.

Proving in ZK that $\mathbf{z}_i^* = \mathrm{ext}(j_i, \mathbf{v}_i^*)$ and $\mathbf{y}_i^* = \mathrm{ext}(\bar{j}_i, \mathbf{w}_i^*)$

1. For $b \in \{0,1\}$, for $\pi \in \mathcal{S}_m$, we define the permutation $F_{b,\pi}$ that transforms vector $\mathbf{z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \end{pmatrix} \in \mathbb{Z}_q^{2m}$ to vector $F_{b,\pi}(\mathbf{z}) = \begin{pmatrix} \pi(\mathbf{z}_b) \\ \pi(\mathbf{z}_{\bar{b}}) \end{pmatrix}$.
2. For all b, π, ϕ, we have:
$$\begin{aligned} \mathbf{z}_i^* = \mathrm{ext}(j_i, \mathbf{v}_i^*) &\iff F_{b,\pi}(\mathbf{z}_i^*) = \mathrm{ext}(\, j_i \oplus b,\, \pi(\mathbf{v}_i^*)\,) \\ \mathbf{y}_i^* = \mathrm{ext}(\bar{j}_i, \mathbf{w}_i^*) &\iff F_{\bar{b},\phi}(\mathbf{y}_i^*) = \mathrm{ext}(\, j_i \oplus b\,, \phi(\mathbf{w}_i^*)\,). \end{aligned}$$
3. $j_i \oplus b$ perfectly hides j_i, if b is a random bit.

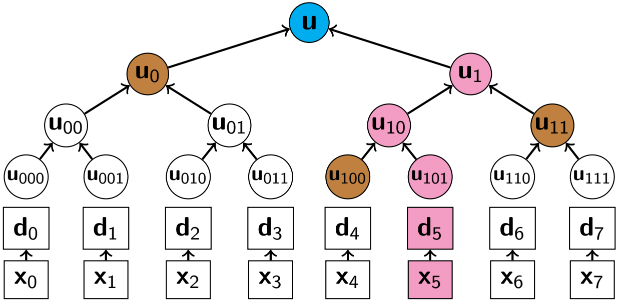
Summary of Our ZK Argument

Putting everything together, in the framework of Stern's protocol, we obtain a ZK argument system for our accumulator.

- When extending the secret vectors, we also extend the public matrices $\mathbf{A}$, $\mathbf{G}$ (by inserting zero-columns) to preserve the equations.
- To prove that the same $\mathbf{v}_i$ is "nested" in 2 equations, we use the same permutation at both places.
- Each round has communication cost $\widetilde{\mathcal{O}}(\ell \cdot n) = \widetilde{\mathcal{O}}(\log N \cdot n)$.
- Each round has soundness error $2/3$, which can be made negligible by repeating $\kappa = \omega(\log n)$ times in parallel.

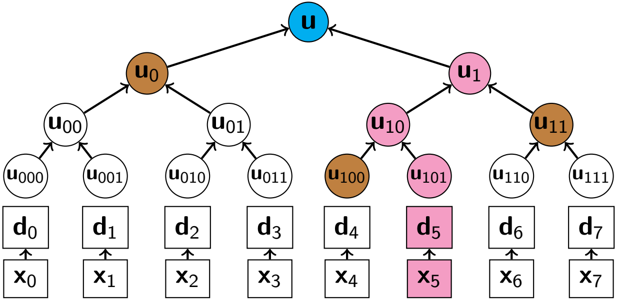
Outline

From ZK-for-Accumulator to Ring Signatures



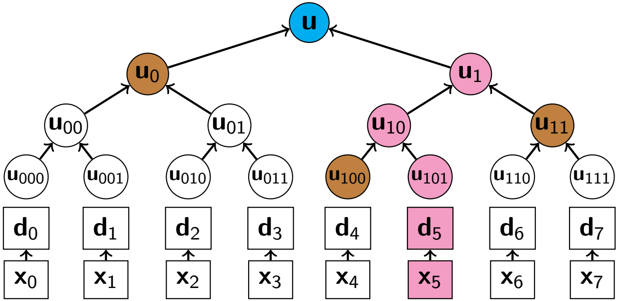
- One more hashing layer is added: Each user picks $sk = \mathbf{x} \xleftarrow{\$} \{0,1\}^m$, and outputs $pk = \mathbf{d} = \mathrm{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{nk}$.

From ZK-for-Accumulator to Ring Signatures



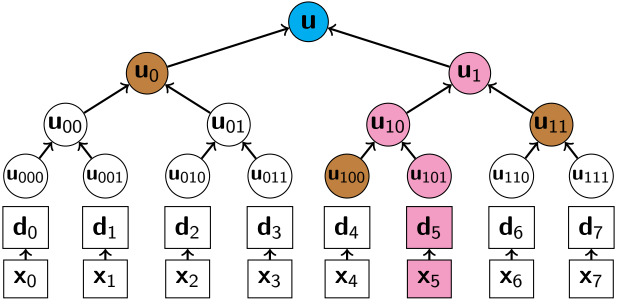
- One more hashing layer is added: Each user picks $sk = \mathbf{x} \xleftarrow{\$} \{0,1\}^m$, and outputs $pk = \mathbf{d} = \mathrm{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{nk}$.
- Signing w.r.t. a ring $R = (pk_0, \ldots, pk_{N-1})$ using $sk = \mathbf{x}$ s.t. $pk \in R$:
  1. Accumulate R into $\mathbf{u}$.
  2. Extend the ZK-argument-for-accumulator to additionally prove knowledge of $\mathbf{x}$ s.t. the value at the secret leaf is $\mathrm{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q)$.
  3. The argument is transformed into a signature via Fiat-Shamir.

From ZK-for-Accumulator to Ring Signatures



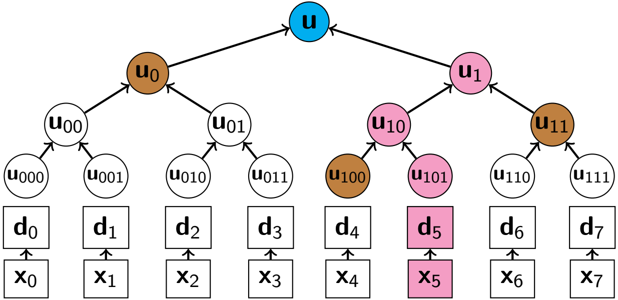
- One more hashing layer is added: Each user picks $sk = \mathbf{x} \xleftarrow{\$} \{0,1\}^m$, and outputs $pk = \mathbf{d} = \mathrm{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q) \in \{0,1\}^{nk}$.
- Signing w.r.t. a ring $R = (pk_0, \ldots, pk_{N-1})$ using $sk = \mathbf{x}$ s.t. $pk \in R$:
  1. Accumulate R into $\mathbf{u}$.
  2. Extend the ZK-argument-for-accumulator to additionally prove knowledge of $\mathbf{x}$ s.t. the value at the secret leaf is $\mathrm{bin}(\mathbf{A} \cdot \mathbf{x} \bmod q)$.
  3. The argument is transformed into a signature via Fiat-Shamir.

From Ring Signatures to Group Signatures



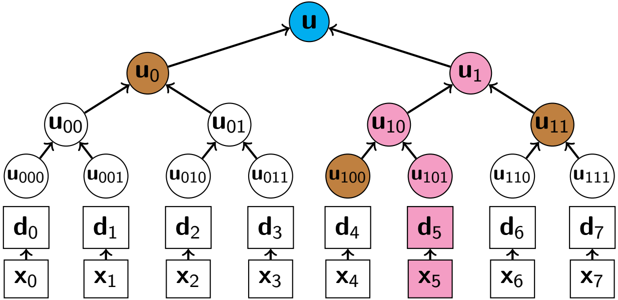
- Fix $N = 2^\ell$. The manager samples $\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}$, computes $\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}$ and the accumulator $\mathbf{u}$. The sk of user j is $\mathbf{x}_j$ and the witness for $\mathbf{d}_j$.

From Ring Signatures to Group Signatures



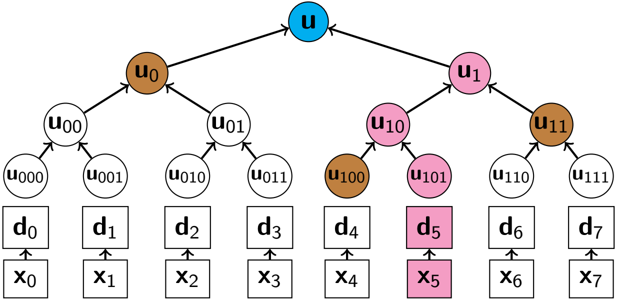
- Fix $N = 2^\ell$. The manager samples $\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}$, computes $\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}$ and the accumulator $\mathbf{u}$. The sk of user j is $\mathbf{x}_j$ and the witness for $\mathbf{d}_j$.
- A CCA-secure encryption layer is added to enable tracing: When signing messages, user j also encrypts the bin. rep. $(j_1, \ldots, j_\ell)$ of j.

From Ring Signatures to Group Signatures



- Fix $N = 2^\ell$. The manager samples $\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}$, computes $\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}$ and the accumulator $\mathbf{u}$. The sk of user j is $\mathbf{x}_j$ and the witness for $\mathbf{d}_j$.
- A CCA-secure encryption layer is added to enable tracing: When signing messages, user j also encrypts the bin. rep. $(j_1, \ldots, j_\ell)$ of j.
- To be trapdoor-less: Use the Naor-Yung double-encryption paradigm [NY'90] with the multi-bit version of Regev's LWE-based encryption [Reg'05].

From Ring Signatures to Group Signatures



- Fix $N = 2^\ell$. The manager samples $\mathbf{x}_0, \ldots, \mathbf{x}_{N-1}$, computes $\mathbf{d}_0, \ldots, \mathbf{d}_{N-1}$ and the accumulator $\mathbf{u}$. The sk of user j is $\mathbf{x}_j$ and the witness for $\mathbf{d}_j$.
- A CCA-secure encryption layer is added to enable tracing: When signing messages, user j also encrypts the bin. rep. $(j_1, \ldots, j_\ell)$ of j.
- To be trapdoor-less: Use the Naor-Yung double-encryption paradigm [NY'90] with the multi-bit version of Regev's LWE-based encryption [Reg'05].
- The argument system for the ring signature is extended to additionally prove that the two ciphertexts correspond to the same plaintext $(j_1, \ldots, j_\ell)$.

Summary

We propose:

- A Merkle-tree-style lattice-based accumulator, supported by short zero-knowledge argument.
- The first lattice-based RS with logarithmic-size signatures.
- The first lattice-based GS without trapdoors.
  Also the first logarithmic-size GS in the [BMW'03] model that does not use a full-fledged digital signature for generating group members' private keys.

Thank you!