Group Signatures and More from Isogenies and Lattices: Generic, Simple, and Efficient

Yi-Fu Lai [2] Joint work with

Ward Beullens[1] , Samuel Dobson[2],

Shuichi Katsumata[3], Federico Pintore[4]

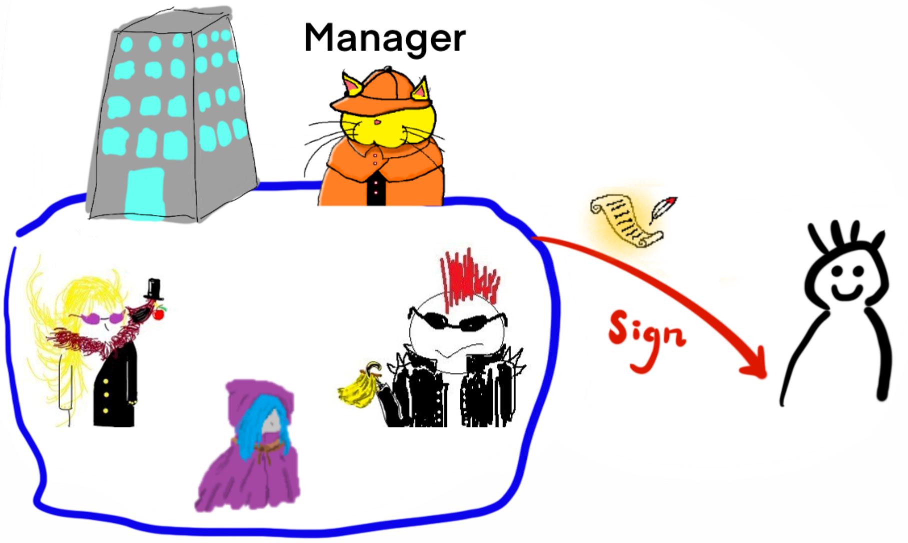
Eurocrypt2022

[1] IBM Research, [2] University of Auckland, [3] AIST, [4] University of Bari

Content

Content

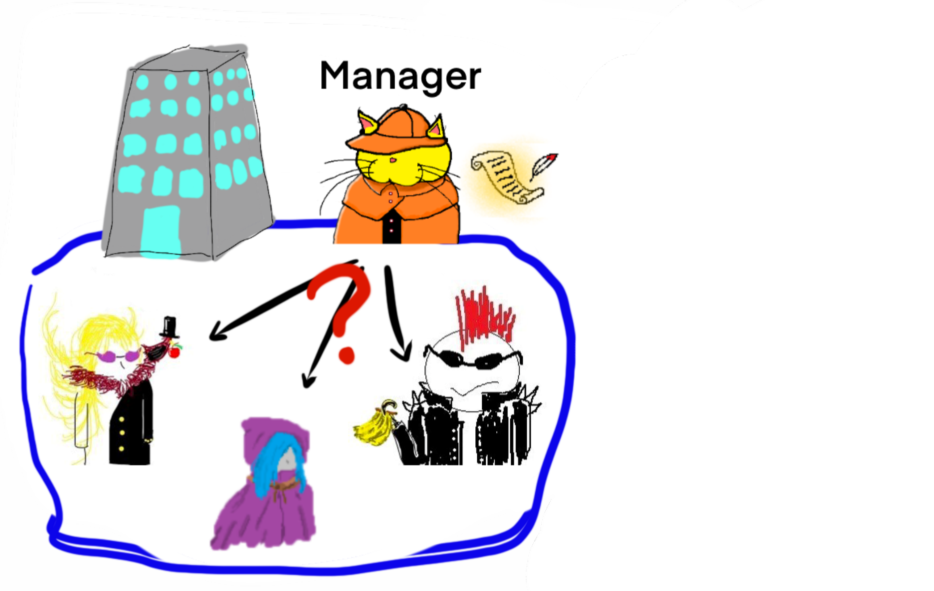
Group Signatures (GS)



Intuitively, a group signature requires

1. Any member in the group can sign anonymously for the group.

Group Signatures (GS)



Intuitively, a group signature requires

1. Any member in the group can sign anonymously for the group.
2. In case of abuse, there is a manager (opener) who can open any signature from the group and know who is the signer and provides a proof.

Security Notions

The requirements for an GS [1]:

1. **CCA (resp. CPA) Anonymity:** Given a signature from any two people chosen by the adversary (resp. withiout access to the opening oracle), it's impossible to tell from which of the two.

2. (**Full**) **Unforgeability:** Any colluding members (with the opener) cannot forge a signature not tracing to one of them.

3. **Traceability:** A valid signature should be able to be opened to one and only one user in the group.

[1] Equivalent to [BSZ05.]

Brief History

- Firstly proposed by Chaum and van Heyst [CV91] by using RSA or DLP assumptions.
- It is formalized in [BMW03,BSZ05] provided with frameworks using verifiable IND-CCA PKE + signature schemes (sign-and-encrypt paradigm).
- Applications and real-world deployments: e.g. directed anonymous attestation and enhanced privacy ID ([BCC04,BL07]), also in a variety of the blockchain and cryptocurrency studies.
- Post-Quantum Proposals: LLLS13, ELL^{+}15, LLNW16, LNWX18, KY19 etc.
- Recently, several proposals have achieved logarithmic property [BCN18, dLS18, EZS^{+}19, ESZ22] where the signature size is logarithmic in the number of the members.

A Question

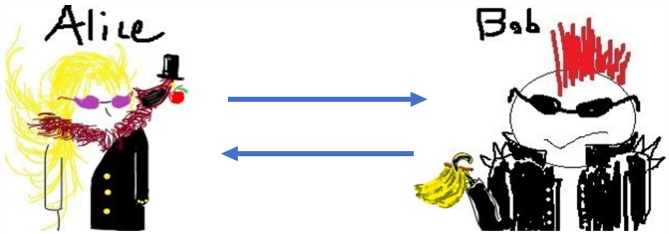
Can we have an isogeny group signature competitive among the post-quantum proposals?

Difficulties

- **CCA-Anonymity:** The standard sign-and-encrypt technique requires IND-CCA verifiable encryption scheme (PKE) because we use
  1. verifiability + signature scheme → unforgeability
  2. the decryption oracle (IND-CCA) to answer the opening oracle queries for CCA anonymity.
- **Full Unforgeability and Traceability:** requires NIZK for the ciphertext and the plaintext.

Difficulties

However, no such practical tools in isogenies with the standard assumptions.



$$\mathbf{ct} = \boldsymbol{H}(j(E_{shared})) \oplus m$$

- **Solutions:** We construct a new verifiable IND-CPA PKE with online-extractable NIZK (but weakly decryptable).

Contributions (Brief)

1. A new practical framework for GS based on group actions with isogeny and lattice instantiations.
2. Logarithmic signature size.
3. Tightly secure variants for the two instantiations.
4. The first GS from isogenies and the only logarithmic one.
5. The isogeny instantiation has the smallest signature size in the literature.

Isogeny Instantiation

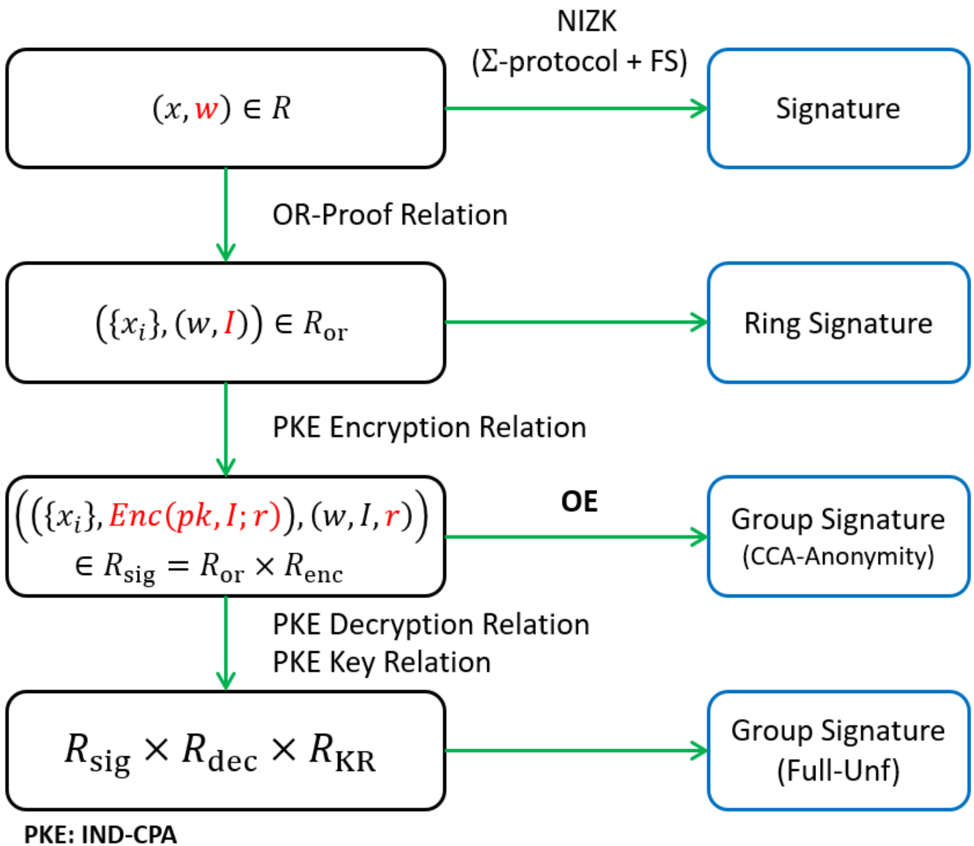
Comparison with other isogeny-based group signature proposals.

Notions	Signature Size	Anonymity	Manager Accountable
[LD21]	$\mathcal{O}(N\log(N))$	CPA	No
[CHH+21]	$\mathcal{O}(N^2)$	CPA	Partially
This Work	$\mathcal{O}(\log(N))$	CCA	Yes

- N: number of members.
- Manager Accountablility: Manager cannot frame an honest member.

Content

Super High Level Idea



Sigma Protocols

Let R be a relation and $(X, W) \in R$. A sigma protocol (Σ-protocol) for R is a three-move interactive protocol

$$\Pi_\Sigma = (P = (P_1, P_2), V = (V_1, V_2))$$

between a prover P with (X, W) and a verifier V with X.

$$\begin{array}{lcl} \textbf{Prover} & & \textbf{Verifier} \\ \mathrm{com} \leftarrow P_1(X, W) & \xrightarrow{\mathrm{com}} & \mathrm{ch} \leftarrow V_1(\mathrm{com}) \\ & \xleftarrow{\mathrm{ch}} & \\ \mathrm{rsp} \leftarrow P_2(W, \mathrm{ch}) & \xrightarrow{\mathrm{rsp}} & \\ & \xleftarrow{1/0} & 1/0 \leftarrow V_2(\mathrm{com}, \mathrm{ch}, \mathrm{rsp}) \end{array}$$

Requirements:

- Correctness
- Special Soundness
- Honest Verifier Zero-knowledge (HVZK)

Group Actions

A group G acts on a set $\mathcal{X}$ by an action $\star : G \times \mathcal{X} \to \mathcal{X}$ if

1. Identity: $\star(e, x) = x$
2. Compatibility: $\star(g, \star(h, x)) = \star(gh, x)$

Abbreviate $\star(g, x)$ as $g \star x$.

Group Actions

A group G acts on a set $\mathcal{X}$ by an action $\star : G \times \mathcal{X} \to \mathcal{X}$ if

1. Identity: $\star(e, x) = x$
2. Compatibility: $\star(g, \star(h, x)) = \star(gh, x)$

Abbreviate $\star(g, x)$ as $g \star x$.

- **Hardness:** given $\star$, $g \star x$ and x, it's hard to recover g.

Group Actions

A group G acts on a set $\mathcal{X}$ by an action $\star : G \times \mathcal{X} \to \mathcal{X}$ if

1. Identity: $\star(e, x) = x$
2. Compatibility: $\star(g, \star(h, x)) = \star(gh, x)$

Abbreviate $\star(g, x)$ as $g \star x$.

- **Hardness:** given $\star$, $g \star x$ and x, it's hard to recover g.

Example

Let n be a natural number, $G = \mathbb{Z}_n$, and $\mathcal{X}$ a cyclic group of order n.
Define $g \star x := x^g$.
The hardness here is based on the discrete logarithm problem over $\mathcal{X}$.

Isogeny Instantiation (CSIDH)

CSIDH ([CLM$^+$18,BKV19]) gives an ideal class group G and a set of supersingular curves $\mathcal{X} = \mathsf{E}_p(\mathcal{O}, \pi)$ such that

- G acts on $\mathcal{X}$ (freely and transitively),
- $E_0 \in \mathcal{X}$.[2]

[2] $E_0 : y^2 = x^3 + x$

Isogeny Instantiation (CSIDH)

CSIDH ([CLM+18,BKV19]) gives an ideal class group G and a set of supersingular curves $\mathcal{X} = \mathrm{E}_p(\mathcal{O}, \pi)$ such that

- G acts on $\mathcal{X}$ (freely and transitively),
- $E_0 \in \mathcal{X}$.[2]

GAIP Problem

Let $s \leftarrow G$. Given $E = s \star E_0$, it's hard to recover $s \in G$.

[2] $E_0 : y^2 = x^3 + x$

Group-Action-Based PKE (GAPKE)

$$\Pi_{\mathsf{PKE}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$$

There are two groups G, G_{M} both acting on a set $\mathcal{X}$.

G_{M} containing the message space $\mathcal{M}$ acts on $\mathcal{X}$ by a public action $\star_{\mathsf{M}}$.

- KeyGen gives $\mathsf{pk} = (\star_{\mathsf{pk}}, X_{\mathsf{pk}})$ and $\mathsf{sk} \leftarrow G$.

Group-Action-Based PKE (GAPKE)

$$\Pi_{\mathsf{PKE}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$$

There are two groups G, G_{M} both acting on a set $\mathcal{X}$.

G_{M} containing the message space $\mathcal{M}$ acts on $\mathcal{X}$ by a public action $\star_{\mathsf{M}}$.

- KeyGen gives $\mathsf{pk} = (\star_{\mathsf{pk}}, X_{\mathsf{pk}})$ and $\mathsf{sk} \leftarrow G$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{M}; r)$ returns the ciphertext $\mathsf{ct} = \mathsf{M} \star_{\mathsf{M}} (r \star_{\mathsf{pk}} X_{\mathsf{pk}}) \in \mathcal{X}$ with a randomness $r \in G$.

Group-Action-Based PKE (GAPKE)

$$\Pi_{\mathsf{PKE}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$$

There are two groups G, G_{M} both acting on a set $\mathcal{X}$.

G_{M} containing the message space $\mathcal{M}$ acts on $\mathcal{X}$ by a public action $\star_{\mathsf{M}}$.

- KeyGen gives $\mathsf{pk} = (\star_{\mathsf{pk}}, X_{\mathsf{pk}})$ and $\mathsf{sk} \leftarrow G$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{M}; r)$ returns the ciphertext $\mathsf{ct} = \mathsf{M} \star_{\mathsf{M}} (r \star_{\mathsf{pk}} X_{\mathsf{pk}}) \in \mathcal{X}$ with a randomness $r \in G$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ is not specified.

Group-Action-Based PKE (GAPKE)

$$\Pi_{\mathsf{PKE}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$$

There are two groups G, G_{M} both acting on a set $\mathcal{X}$.

G_{M} containing the message space $\mathcal{M}$ acts on $\mathcal{X}$ by a public action $\star_{\mathsf{M}}$.

- KeyGen gives $\mathsf{pk} = (\star_{\mathsf{pk}}, X_{\mathsf{pk}})$ and $\mathsf{sk} \leftarrow G$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{M}; r)$ returns the ciphertext $\mathsf{ct} = \mathsf{M} \star_{\mathsf{M}} (r \star_{\mathsf{pk}} X_{\mathsf{pk}}) \in \mathcal{X}$ with a randomness $r \in G$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ is not specified.
- We assume the PKE scheme is IND-CPA.

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.
- $\mathcal{M} \subset G = G_{\mathsf{M}}$ is **small**.
- $\mathcal{X} = (\mathsf{E}_p(\mathcal{O}, \pi))^2$.

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.
- $\mathcal{M} \subset G = G_{\mathsf{M}}$ is **small**.
- $\mathcal{X} = (\mathsf{E}_p(\mathcal{O}, \pi))^2$.
- KeyGen outputs sk $\leftarrow G$ and pk $= (E_0, X_{\mathsf{pk}} = (E_0, \mathsf{sk} \star E_0))$.

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.
- $\mathcal{M} \subset G = G_{\mathsf{M}}$ is **small**.
- $\mathcal{X} = (\mathsf{E}_p(\mathcal{O}, \pi))^2$.
- KeyGen outputs sk $\leftarrow G$ and pk $= (E_0, X_{\mathsf{pk}} = (E_0, \mathsf{sk} \star E_0))$.
- We have an Elgamal-type encryption[3]:

$$\mathsf{ct} = (r \star E_0, (r + m) \star E_{\mathsf{pk}}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r \leftarrow G).$$

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.
- $\mathcal{M} \subset G = G_{\mathsf{M}}$ is **small**.
- $\mathcal{X} = (\mathsf{E}_p(\mathcal{O}, \pi))^2$.
- KeyGen outputs sk $\leftarrow G$ and pk $= (E_0, X_{\mathsf{pk}} = (E_0, \mathsf{sk} \star E_0))$.
- We have an Elgamal-type encryption[3]:

$$\mathsf{ct} = (r \star E_0, (r + m) \star E_{\mathsf{pk}}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r \leftarrow G).$$

- The decryption of ct $= (E_1, E_2)$ with sk returns m' by **enumerating** elements in $\mathcal{M}$ s.t. $(m' + \mathsf{sk}) \star E_1 = E_2$. Otherwise, it returns $\perp$.

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Isogeny Instantiations (GAPKE)

- Recall G acts on $\mathsf{E}_p(\mathcal{O}, \pi)$ by $\star$ from CSIDH.
- $\mathcal{M} \subset G = G_{\mathsf{M}}$ is **small**.
- $\mathcal{X} = (\mathsf{E}_p(\mathcal{O}, \pi))^2$.
- KeyGen outputs $\mathsf{sk} \leftarrow G$ and $\mathsf{pk} = (E_0, X_{\mathsf{pk}} = (E_0, \mathsf{sk} \star E_0))$.
- We have an Elgamal-type encryption[3]:

$$\mathsf{ct} = (r \star E_0, (r + m) \star E_{\mathsf{pk}}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r \leftarrow G).$$

- The decryption of $\mathsf{ct} = (E_1, E_2)$ with sk returns m' by **enumerating** elements in $\mathcal{M}$ s.t. $(m' + \mathsf{sk}) \star E_1 = E_2$. Otherwise, it returns $\perp$.

Decisional CSIDH Problem

Let $a, b \leftarrow G$. Given $(E_0, a \star E_0, b \star E_0, E)$, where E is either $(a + b) \star E_0$ or $E = c \star E_0$ for some $c \leftarrow G$. It's difficult to distinguish the distribution of E.

[3] $m \star_{\mathsf{M}} (E_1, E_2) := (E_1, m \star E_2)$ and $r \star_{\mathsf{pk}} (E_1, E_2) := (r \star E_1, r \star E_2)$

Content

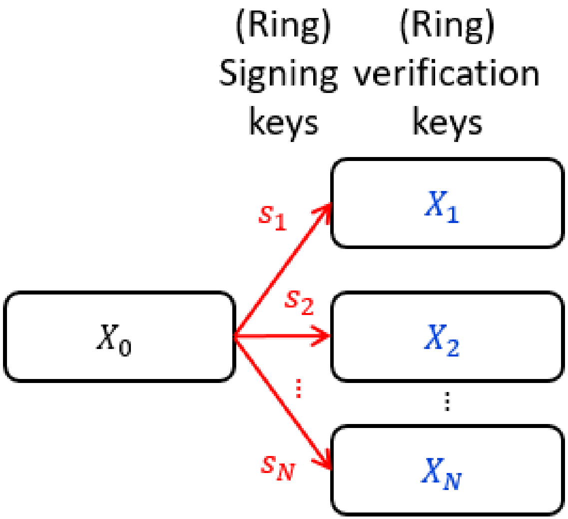
OR-Proof

We start with the relation from [BKP20].

OR-Proof

We start with the relation from [BKP20].

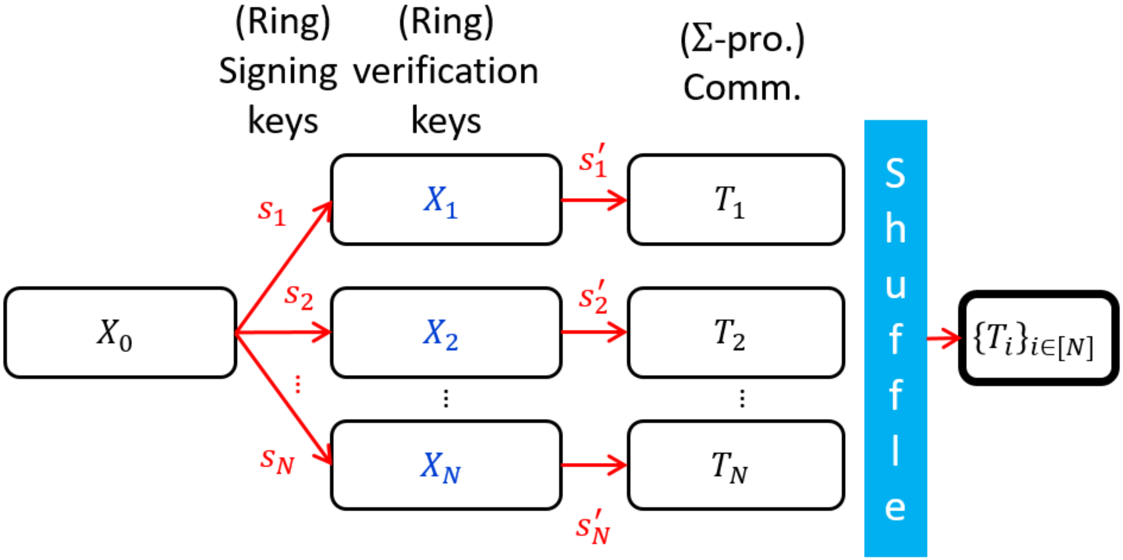
$$R_{\mathrm{or}} = \left\{ \left(\{X_i\}_{i \in [N]}, (s_I, I) \right) \mid s_I \star X_0 = X_I \in \{X_i\}_{i \in [N]} \right\}$$



OR-Proof

We start with the relation from [BKP20].

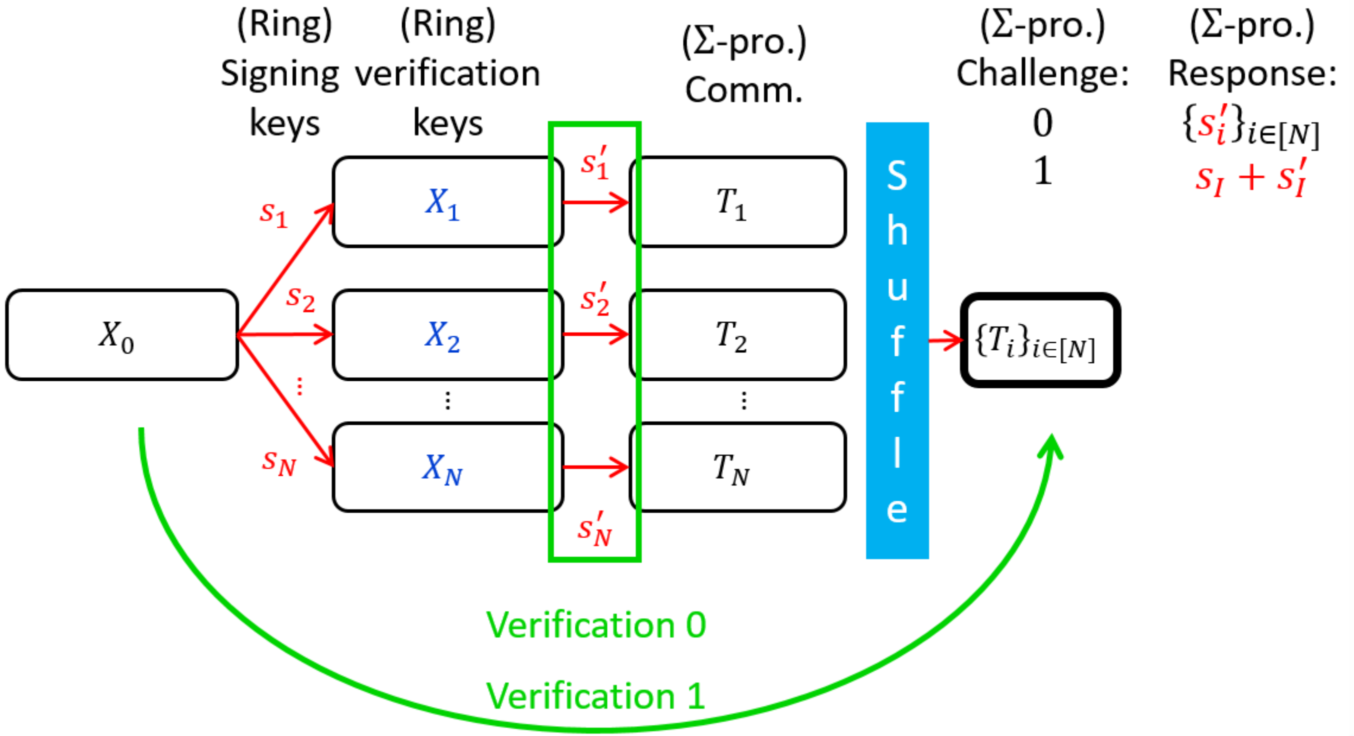
$$R_{\mathrm{or}} = \left\{ \left(\{X_i\}_{i \in [N]}, (s_I, I) \right) \mid s_I \star X_0 = X_I \in \{X_i\}_{i \in [N]} \right\}$$



OR-Proof

We start with the relation from [BKP20].

$$R_{\mathrm{or}} = \left\{ \left(\{X_i\}_{i\in[N]}, (s_I, I) \right) \mid s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \right\}$$



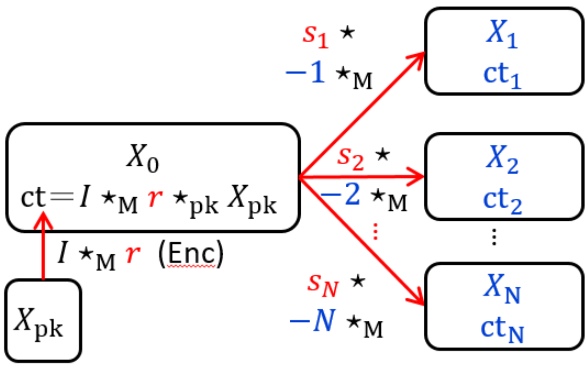
Encryption Relation

The idea here is to concatenate and shuffle two proofs together.

$$R_{\mathrm{or}} \times R_{\mathrm{enc}} = \left\{ \left(\{X_i\}_{i\in[N]}, \mathrm{pk}, \mathrm{ct}, (s_I, I, r) \right) \;\middle|\; \begin{array}{c} s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \\ \mathrm{ct} = \mathrm{Enc}(\mathrm{pk}, I; r) = I \star_{\mathrm{M}} r \star_{\mathrm{pk}} X_{\mathrm{pk}} \end{array} \right\}$$

(Ring) Signing keys

(Ring) verification keys



Encryption Relation

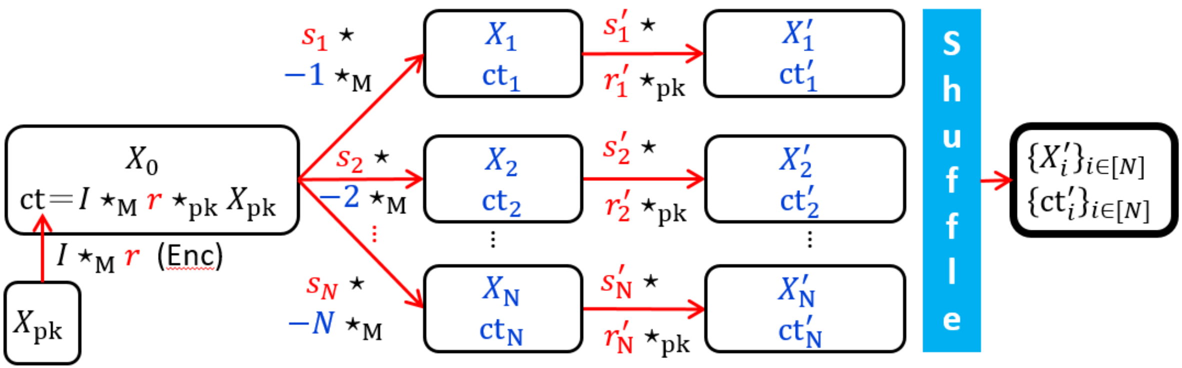
The idea here is to concatenate and shuffle two proofs together.

$$R_{\mathrm{or}} \times R_{\mathrm{enc}} = \left\{ \left(\{X_i\}_{i\in[N]}, \mathrm{pk}, \mathrm{ct}, (s_I, I, r) \right) \;\middle|\; \begin{array}{c} s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \\ \mathrm{ct} = \mathrm{Enc}(\mathrm{pk}, I; r) = I \star_{\mathrm{M}} r \star_{\mathrm{pk}} X_{\mathrm{pk}} \end{array} \right\}$$

(Ring) Signing keys

(Ring) verification keys

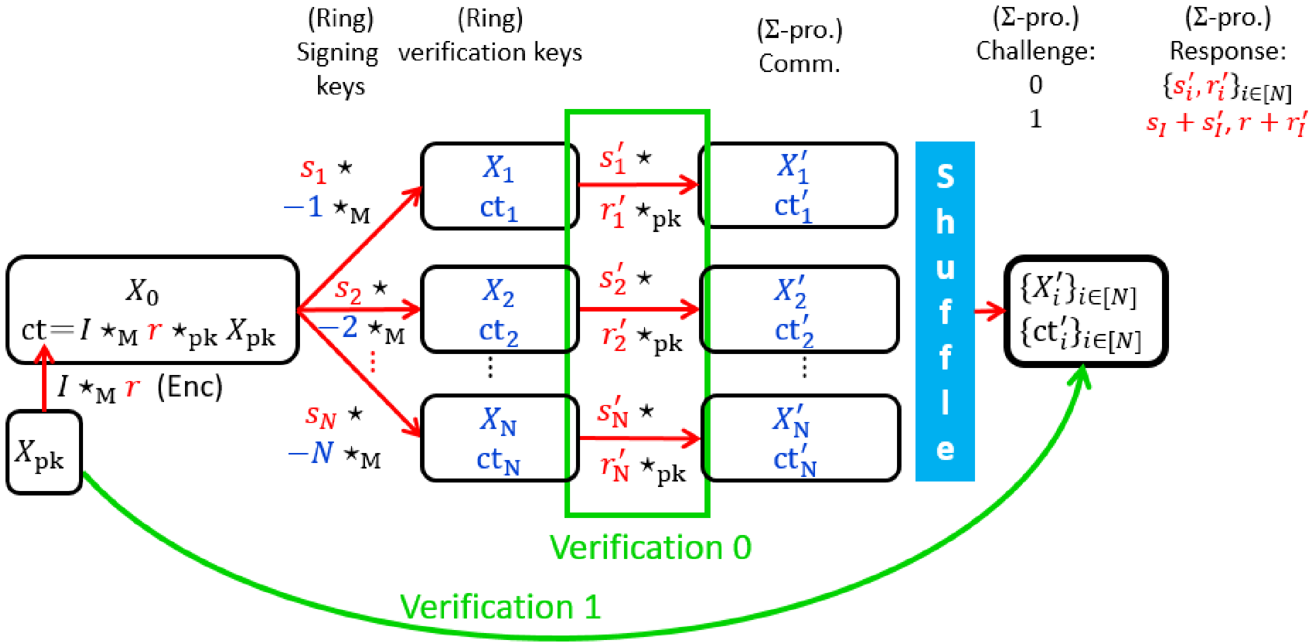
(Σ-pro.) Comm.



Encryption Relation

The idea here is to concatenate and shuffle two proofs together.

$$R_{\text{or}} \times R_{\text{enc}} = \left\{ \left(\{X_i\}_{i \in [N]}, \text{pk}, \text{ct}, (s_I, I, r) \right) \;\middle|\; \begin{array}{c} s_I \star X_0 = X_I \in \{X_i\}_{i \in [N]} \\ \text{ct} = \text{Enc}(\text{pk}, I; r) = I \star_{\text{M}} r \star_{\text{pk}} X_{\text{pk}} \end{array} \right\}$$



Logarithmic Proof

Optimize by using PRNG, Merkle Trees, commitment schemes.

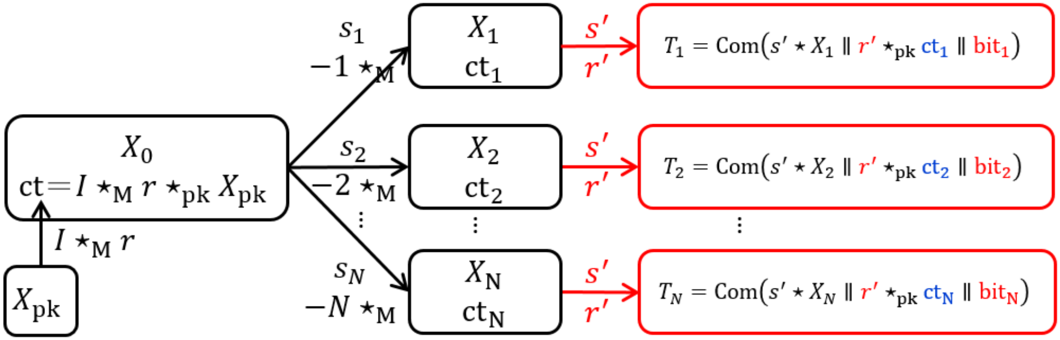
$$R_{\mathrm{or}} \times R_{\mathrm{enc}} = \left\{ \left(\{X_i\}_{i\in[N]}, \mathrm{pk}, \mathrm{ct}, (s_I, I, r) \right) \middle| \begin{array}{c} s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \\ \mathrm{ct} = \mathrm{Enc}(\mathrm{pk}, I; r) = I \star_{\mathrm{M}} r \star_{\mathrm{pk}} X_{\mathrm{pk}} \end{array} \right\}$$

(Ring) Signing keys

(Ring) verification keys

(Σ-pro.) Comm.

(Σ-pro.) Challenge: 0 1

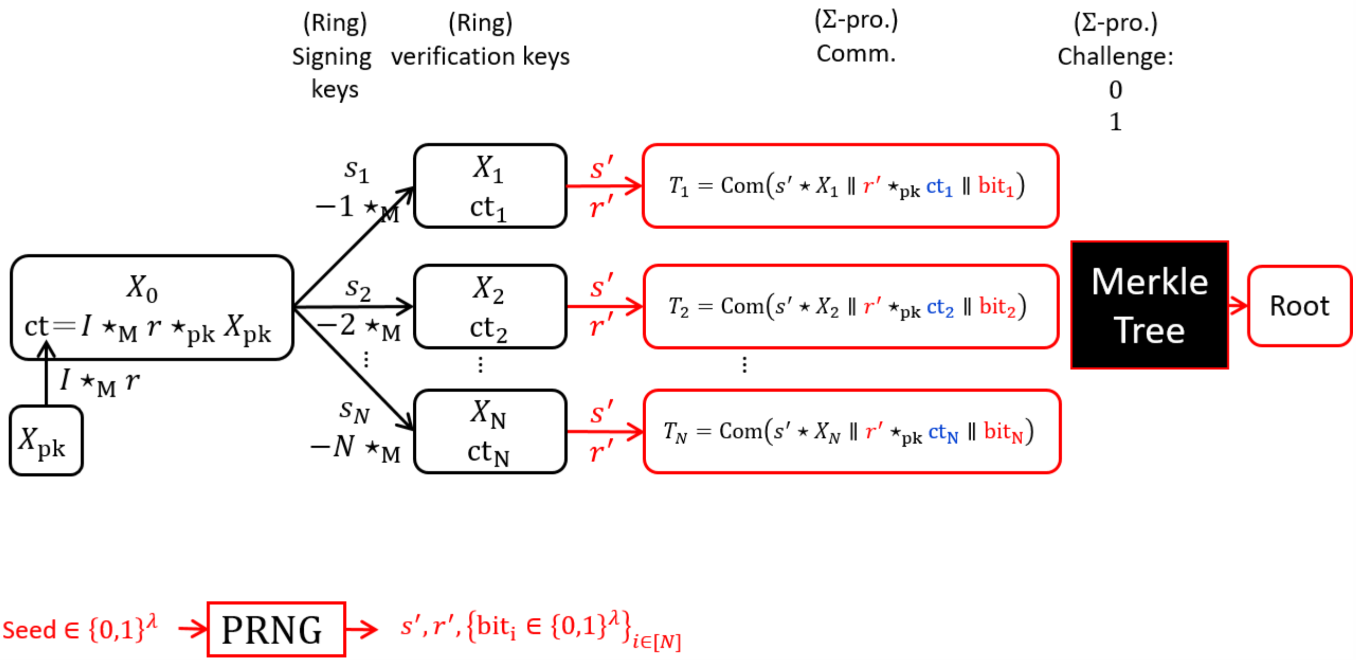


Seed $\in \{0,1\}^{\lambda} \rightarrow \boxed{\mathrm{PRNG}} \rightarrow s', r', \{\mathrm{bit}_i \in \{0,1\}^{\lambda}\}_{i\in[N]}$

Logarithmic Proof

Optimize by using PRNG, Merkle Trees, commitment schemes.

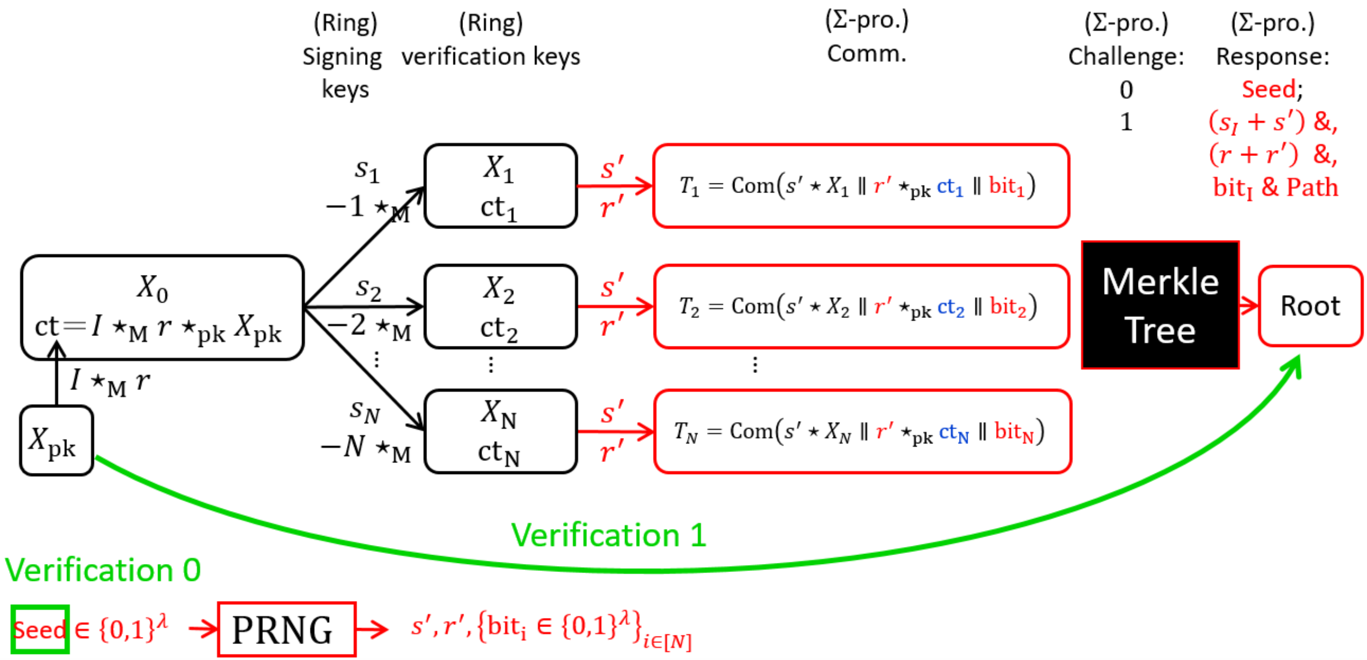
$$R_{\mathrm{or}} \times R_{\mathrm{enc}} = \left\{ \left(\{X_i\}_{i\in[N]}, \mathrm{pk}, \mathrm{ct}, (s_I, I, r) \right) \;\middle|\; \begin{matrix} s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \\ \mathrm{ct} = \mathrm{Enc}(\mathrm{pk}, I; r) = I \star_{\mathrm{M}} r \star_{\mathrm{pk}} X_{\mathrm{pk}} \end{matrix} \right\}$$



Logarithmic Proof

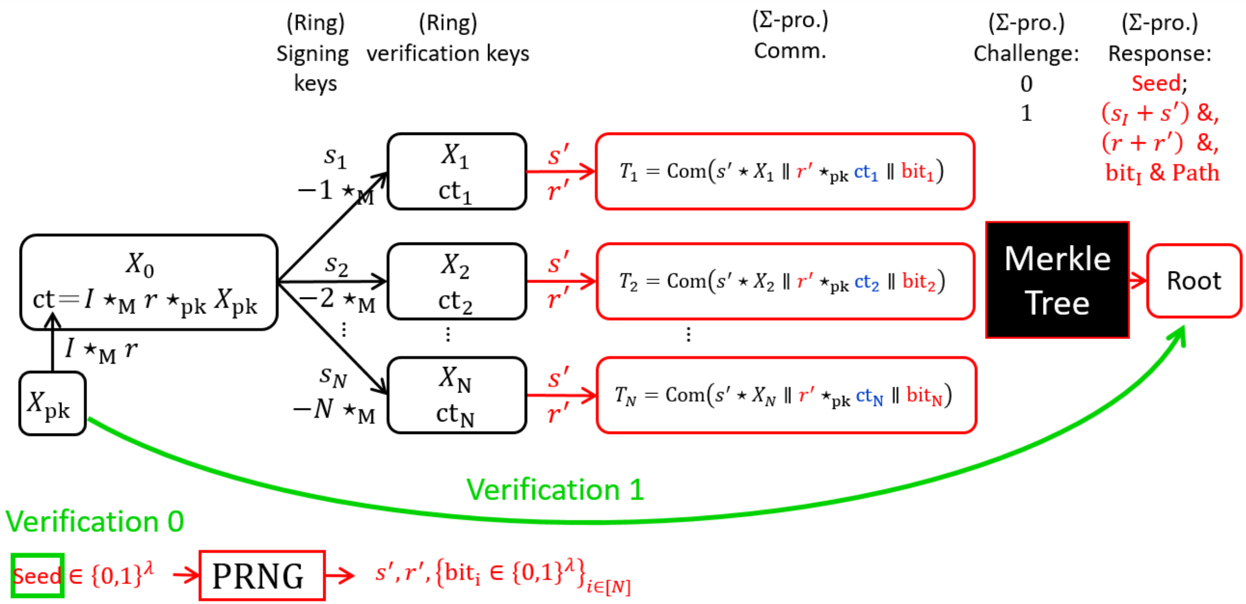
Optimize by using PRNG, Merkle Trees, commitment schemes.

$$R_{\mathrm{or}} \times R_{\mathrm{enc}} = \left\{ \left(\{X_i\}_{i\in[N]}, \mathrm{pk}, \mathrm{ct}, (s_I, I, r) \right) \;\middle|\; \begin{array}{c} s_I \star X_0 = X_I \in \{X_i\}_{i\in[N]} \\ \mathrm{ct} = \mathrm{Enc}(\mathrm{pk}, I; r) = I \star_{\mathrm{M}} r \star_{\mathrm{pk}} X_{\mathrm{pk}} \end{array} \right\}$$



Online-Extractability (OE)

We show OE by modeling PRNG/commitment schemes/Merkle trees as a random oracle.
The main reason is the challenge space size is 2 and one response can be obtained by observing the oracle queries.



"Traceable" Sigma Protocol

Repeat λ times, the interactive protocol will have 2^λ strength.

Via Fiat-Shamir transform, the protocol can be transformed into a non-interactive ring signature of form $(\{X_i\}_{i\in[N]}, \mathsf{pk}, \mathsf{ct}, \sigma)$.

Roughly,

- Online Extractability + IND-CPA $\rightarrow$ CCA anonymity
- Online Extractability + Hardness assumption of the action $\rightarrow$ Unforgeability
- The ciphertext is ct = $\mathsf{Enc}(\mathsf{pk}, I; r) = I \star_{\mathsf{M}} r \star_{\mathsf{pk}} X_{\mathsf{pk}}$. The manager with the decryption key can open the signature.
- It suffices to construct an NIZK for the decryption and key relations (for traceability/full-unf).

The Decryption and Key Validation Relations

By using a similar method, we construct NIZKs for the decryption relations and PKE key relations for our GAPKEs.

- Isogeny:

$$\{((E_0, E_1, E_2, E_3, \mathsf{M}), \mathsf{sk}) \mid E_1 = \mathsf{sk} \star E_0, \mathsf{M} \star \mathsf{sk} \star E_2 = E_3\}.$$

- Lattice:

$$\left\{((\mathbf{A}, \mathbf{e}, \mathbf{b}, \mathbf{c}, c, \ \mathsf{M}), \mathsf{sk} = (\mathbf{s}, \mathbf{z})) \mid \begin{array}{c} \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{z}, \\ \mathbf{s}, \mathbf{z}, c - \mathbf{c}^T\mathbf{s} - \mathsf{M}\lfloor q/2 \rceil \text{ are short.} \end{array}\right\}.$$

- The opener provides the proof for the opening result using NIZK for the relation. Traceability and full-unforgeability will follow.

Other results.

- Reduce the signature size:
  - Using the unbalanced challenge space (#0s>#1s).
- Lattice instantiation:
  - We give GAPKE by using Lindner-Peikert framework [LP11].
  - The signature size can be further reduced by using the Bai-Galbraith method.
- Tightly secure variant:
  - Using the Katz-Wang method.
  - The (unforgeability) reduction loss is only $1/2$. (ϵ^2/N^2 mostly.)
  - The additional cost is only a constant[4].

[4] Increased by 0.5 KB; signing, verification slow down by factor 2.

Content

Result: post-quantum group signatures

Comparison with other post-quantum group signature proposals.

	N					Hardness Assumption	Security Level	Anonymity	Manager Accountable
	2	2^5	2^6	2^{10}	2^{21}				
Isogeny	3.6	6.0	6.6	9.0	15.5	CSIDH-512	$*$	CCA	Yes
Lattice	124	126	126	129	134	MSIS/MLWE	NIST 2	CCA	Yes
Lattice	86	88	89	91	96	MSIS/MLWE	NIST 2	CCA	No
[ESZ22]	/	12	/	19	/	MSIS/MLWE	NIST 2	CPA	No
[KKW18]	/	/	280	418	/	LowMC	NIST 5	selfless-CCA	No

- N: number of memebers. Signature size is in KB.
- *: estimated to be 60 bits of quantum security in [Pei20].
- Non-Selfless: anonymous against full-key exposure.
- Manager Accountablility: Manager cannot frame an honest member.

Contributions

1. A new framework for GS based on group actions with isogeny and lattice instances achieving all ideal security properties specified in [BSZ05].
2. Our framework is logarithmic. Concretely, the size of
   - the isogeny instance has the smallest order of magnitude in the literature (e.g. 6.6 KB for 64 members).
   - the lattice instance has the smallest growth rate in the lattice literature[5].
3. The first two tightly secure post-quantum GS.
4. The first GS from isogenies and the only logarithmic proposal.

[5] $0.5\log_2(N) + 85.9$ KB

Thanks for listening!