30th Selected Areas in Cryptography

Traceable Ring Signatures from Group Actions: Logarithmic, Flexible, and Quantum Resistant

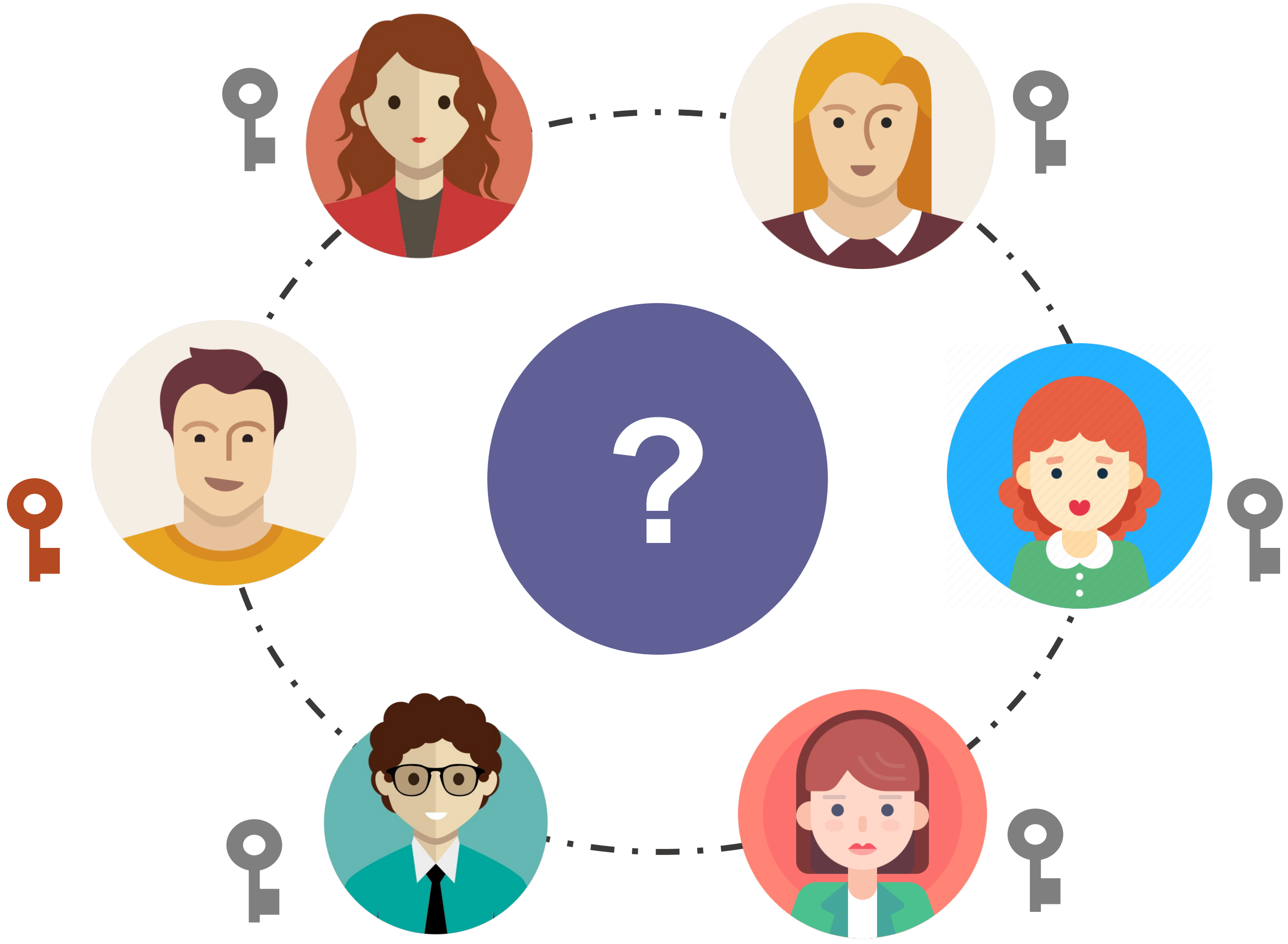
Reporter: Wei Wei

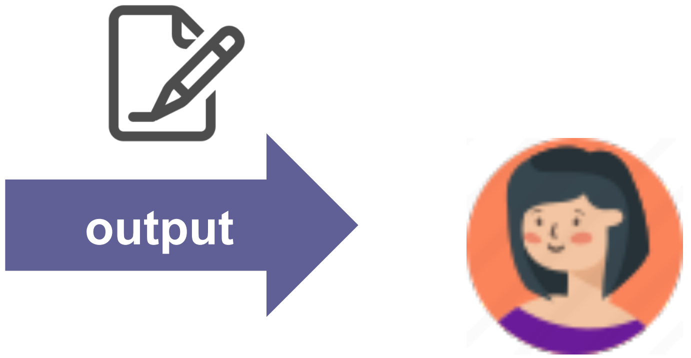
Authors: **Wei Wei**, Min Luo*, Zijian Bao, Cong Peng and Debiao He*

Wuhan University

2023-08-16

Motivation — Ring Signature





Hides the origin of a signature

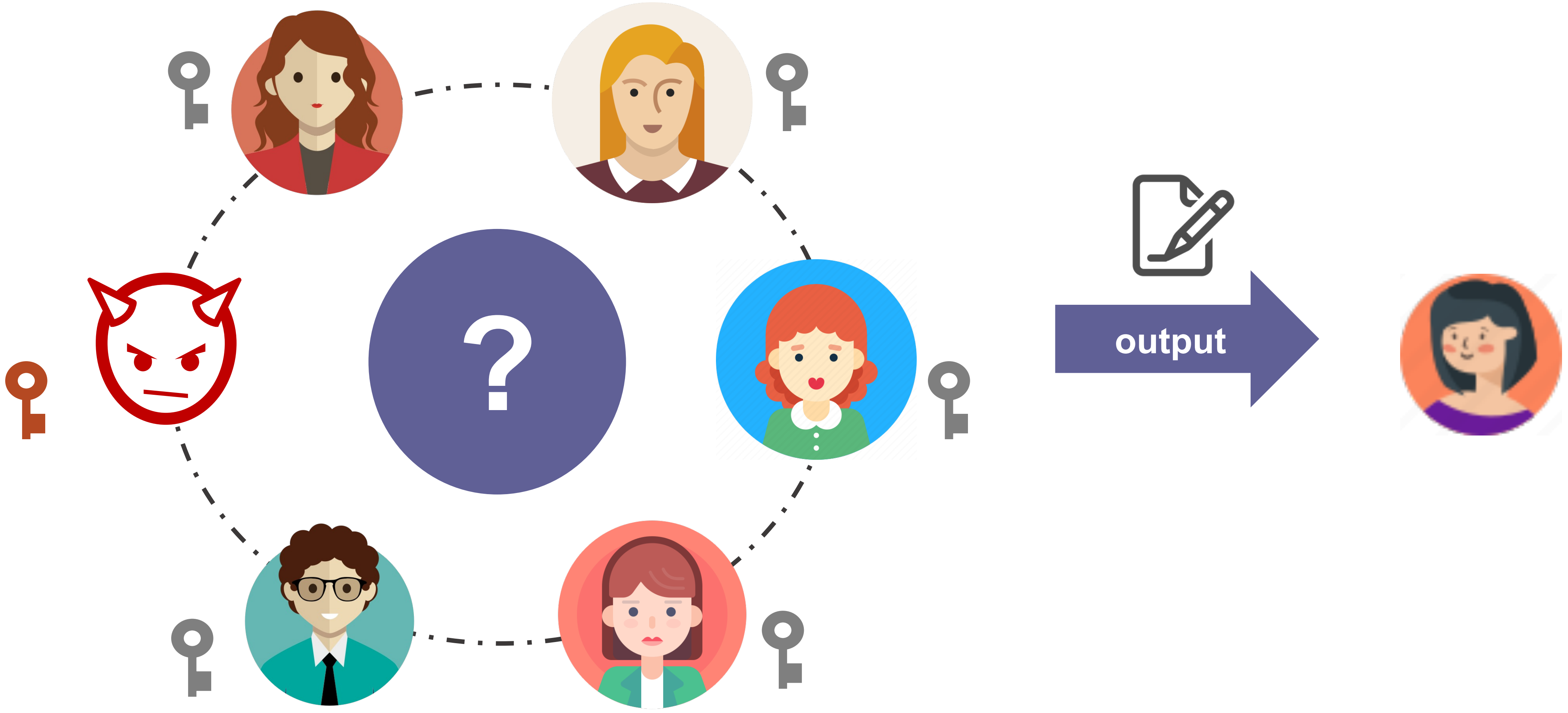
Protect the privacy of signers

Motivation — Ring Signature

e-voting

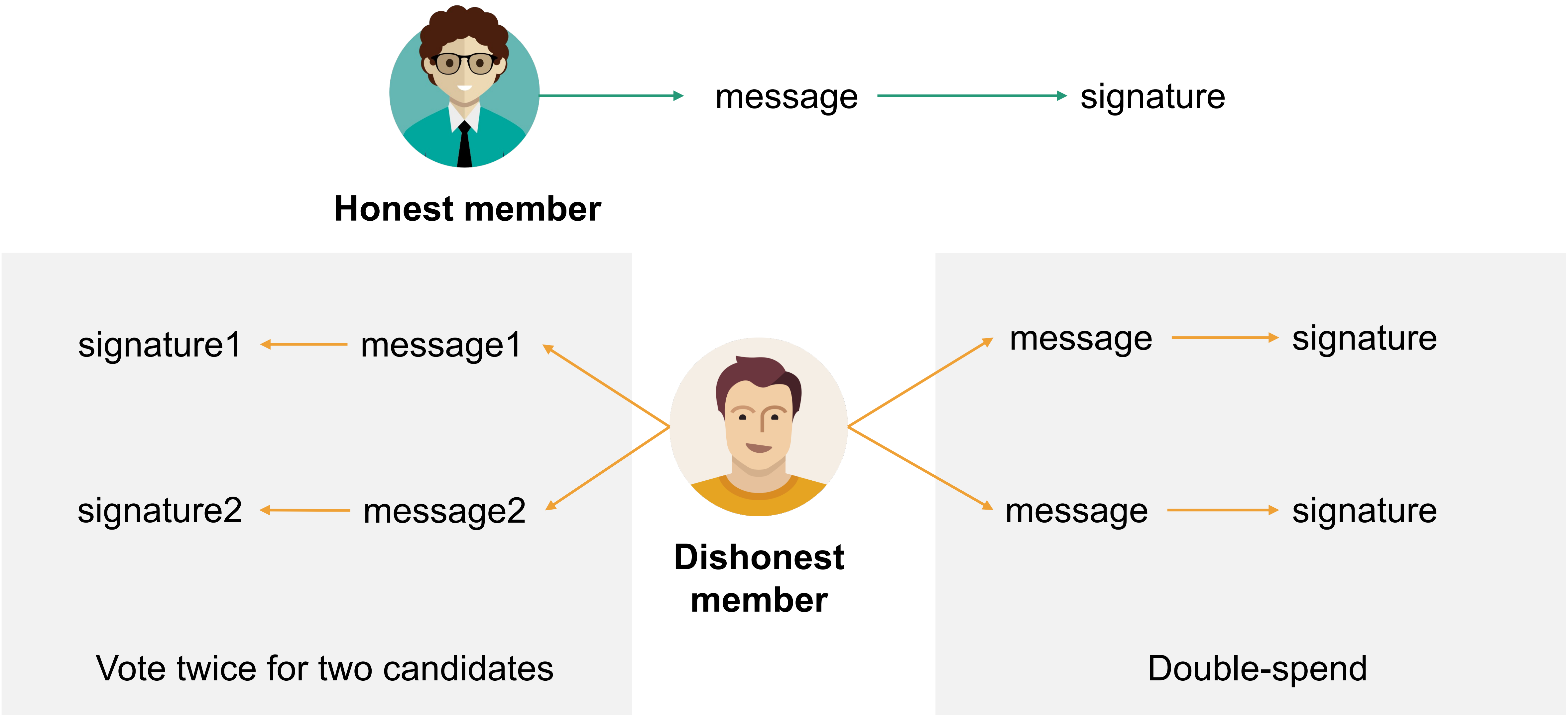
e-coupon services

Motivation — Ring Signature



Unconditional anonymity

Motivation — Traceable Ring Signature



- **How to construct a post-quantum secure traceable ring signature?**

Motivation — Literature review

- Lattice-based schemes

  ring signature
  traceable ring signature
  unique ring signature

- Isogeny-based schemes

  linkable ring signature
  accountable ring signature
  revocable ring signature

- Other post-quantum schemes

  traceable ring signature
  one-time traceable ring signature

Table 1: Comparison of our TRS with other (traceable) ring signature.

Schemes	Signature size	Linkability	Traceability	Implementation	Hardness Assumption
Alessandra[32]	$O(N)$	✓	✓	✓	NONE
Branco[6]	$O(N)$	✓	✓	×	SD[1]
Falafl[4]	$O(\log(N))$	✓	×	✓	MSIS[2], MLWE[3]
Feng H[19]	$O(\log(N))$	✓	✓	×	SIS [2], LWE[3]
MatRiCT[18]	$O(\log(N))$	×	×	✓	MSIS[2], MLWE[3]
Esgin[16]	$O(\log(N))$	×	×	✓	SIS[2], LWE[3]
Raptor[27]	$O(N)$	✓	×	✓	NTRU[4]
Calamari[4]	$O(\log(N))$	✓	×	✓	CSIDH[5]
CHH[10]	$O(N^2)$	×	✓	×	CSIDH[5]
KYM[24]	$O(N\log(N))$	×	✓	×	CSIDH[5]
This work	$O(\log(N))$	✓	✓	✓	MSIS[2], MLWE[3], CSIDH[5]

[1] SD: Syndrome Decoding
[2] SIS: Short Integer Solution, MSIS: Module Short Integer Solution
[3] LWE: Learning with Errors, MLWE: Module Learning with Errors
[4] NTRU: Number Theory Research Unit
[5] CSIDH: Commutative Supersingular Isogeny Diffie Hellman

Motivation — Literature review

[BKP2020]

- construct an efficient **(linkable) ring signature** scheme and gave two concrete instances from isogenies and lattices
- a general OR-proof, logarithmic signature size

[BKP2023]

- construct an efficient **dynamic group signature** (accountable ring signature) from isogeny and lattice assumptions
- add a proof of valid ciphertext to [BKP2020]'s OR-proof and proving full anonymity

- **Is this the construction for other signature schemes (traceable ring signature)?**

Background — Restricted Pair of Group Action

Group Action

Let $(\mathcal{G},\cdot)$ be a group with identity element $\mathfrak{e} \in \mathcal{G}$ and $\mathcal{X}$ a set. A map $\star\colon \mathcal{G}\times\mathcal{X} \to \mathcal{X}$ is a group action if it satisfies the following properties:

- Compatibility: $(\mathfrak{g} \cdot \mathfrak{h}) \star x = (\mathfrak{h} \cdot \mathfrak{g}) \star x$ for all $\mathfrak{g}, \mathfrak{h} \in \mathcal{G}$ and $x \in \mathcal{X}$.
- Identity: $\mathfrak{e} \star x = x$ for all $x \in \mathcal{X}$.

Restricted Effective Group Action

Let $(\mathcal{G},\mathcal{X},\star)$ be a group action and let $\vec{\mathfrak{g}} = \{\mathfrak{g}_1, \cdots, \mathfrak{g}_n\}$ be a generating set for G. we call $(\mathcal{G},\mathcal{X},\star,\tilde{x})$ a restricted effective group action if:

1. The group $\mathcal{G}$ is finite and $n = \mathrm{poly}(\log(\#\mathcal{G}))$.
2. membership testing and unique representation.
3. There exists a distinguished element $\tilde{x} \in \mathcal{X}$ with known representation.
4. There exists an efficient algorithm that given $\mathfrak{g}_i \in \mathcal{G}$ and $x \in \mathcal{X}$, outputs $\mathfrak{g}_i \star x$ and $\mathfrak{g}_i^{-1} \star x$.

Background — Restricted Pair of Group Action

Restricted Pair of Effective Group Action

Given a finite commutative group $\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2 \subseteq \mathcal{G}$, $\mathcal{S}$ and $\mathcal{T}$ are two sets. For $(S_0, T_0) \in \mathcal{S} \times \mathcal{T}$, we say that $(\mathcal{G}, \mathcal{S}, \mathcal{T}, \mathcal{G}_1, \mathcal{G}_2)$ is a ξ-restricted pair of group actions if the following holds:

- Efficient Group Action: For any $g \in \mathcal{G}_\mathbf{1} \cup \mathcal{G}_\mathbf{2}$ and $(S, T) \in \mathcal{S} \times \mathcal{T}$, it is efficient to compute $g \star S$ and $g \star T$.
- Efficient Rejection Sampling: For all $g \in \mathcal{G}_1$, the intersection of all sets $\mathcal{G}_\mathbf{2} + g$ is large enough. Let $\mathcal{G}_\mathbf{3} = \bigcap_{g \in \mathcal{G}_1} \mathcal{G}_\mathbf{2} + g$, then $|\mathcal{G}_\mathbf{3}| = \xi |\mathcal{G}_\mathbf{2}|$.
- Efficient Membership Testing: It is efficient to verify that an element $z \in \mathcal{G}_1$, or $z \in \mathcal{G}_2$, or $z \in \mathcal{G}_3$.

- $(g \star S_0, g \star T_0) \approx (S, T)$, s.t. $g \xleftarrow{\$} \mathcal{G}_1, (S, T) \xleftarrow{\$} \mathcal{S} \times \mathcal{T}$ — anonymity
- It is difficult to find $g, g' \in \mathcal{G}_2 + \mathcal{G}_3$, s.t. $g \star S_0 = g' \star S_0$ and $g \star T_0 \neq g' \star T_0$. — tag-linkability
- Given $S = g \star S_0, T = g \star T_0$, it is hard to find $g' \in \mathcal{G}_\mathbf{2} + \mathcal{G}_\mathbf{3}$, s.t. $T = g' \star T_0$ or $S = g' \star S_0$. — exculpability

Construction — Idea

We introduce tag sets to build traceable ring signatures → validity and traceability

- A **general traceable ring signature scheme** is constructed based on OR sigma protocol and group action.
- Each user generates a **tag set** based on message.
- **Traceability** will be possible by checking whether each tag/vector in the two sets is equal.
- **Validity** will be ensured by adding the tag set to OR-proof.
- **Logarithmic signature size under** Isogeny-based and lattice-based Instantiation.

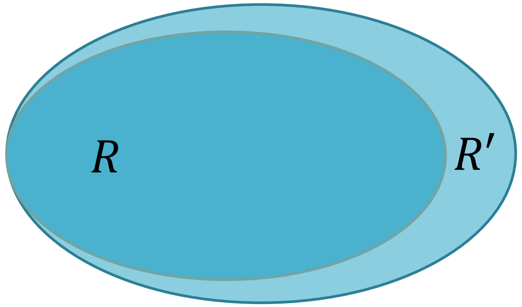
Construction — Definition of relation

- The relation $R \subset \mathcal{S}^{N+1} \times \mathcal{T}^{N+1} \times (\mathcal{G}_1, \mathbb{Z}_N)$

$$R = \{(S_0, S_1, \dots, S_N), (T_0, T_1, \dots, T_N), (g, \pi), \mid g \in \mathcal{G}_1, S_i \in \mathcal{S}, T_i \in \mathcal{T}, S_\pi = g \star S_0, T_\pi = g \star T_0\}$$

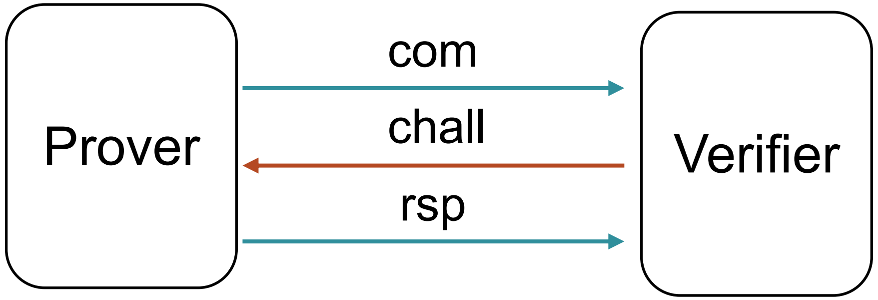
- The relation R' slightly wider than the relation R: $R \subseteq R'$

$$\left\{ (S_0, S_1, \dots, S_N), (T_0, T_1, \dots, T_N), w \,\middle|\, \begin{array}{c} S_i \in \mathcal{S}, T_i \in \mathcal{T} \;\; and \\ w = (g, \pi): g \in \mathcal{G}_1, S_\pi = g \star S_0 \\ T_\pi = g \star T_0 \; or \\ w = (x, x'): x \neq x', H_2(x) = H_2(x') \end{array} \right\}$$



Under (R, R'), the OR sigma protocol is still useful as long as (R, R') is sufficiently difficult.

Construction — OR sigma protocol

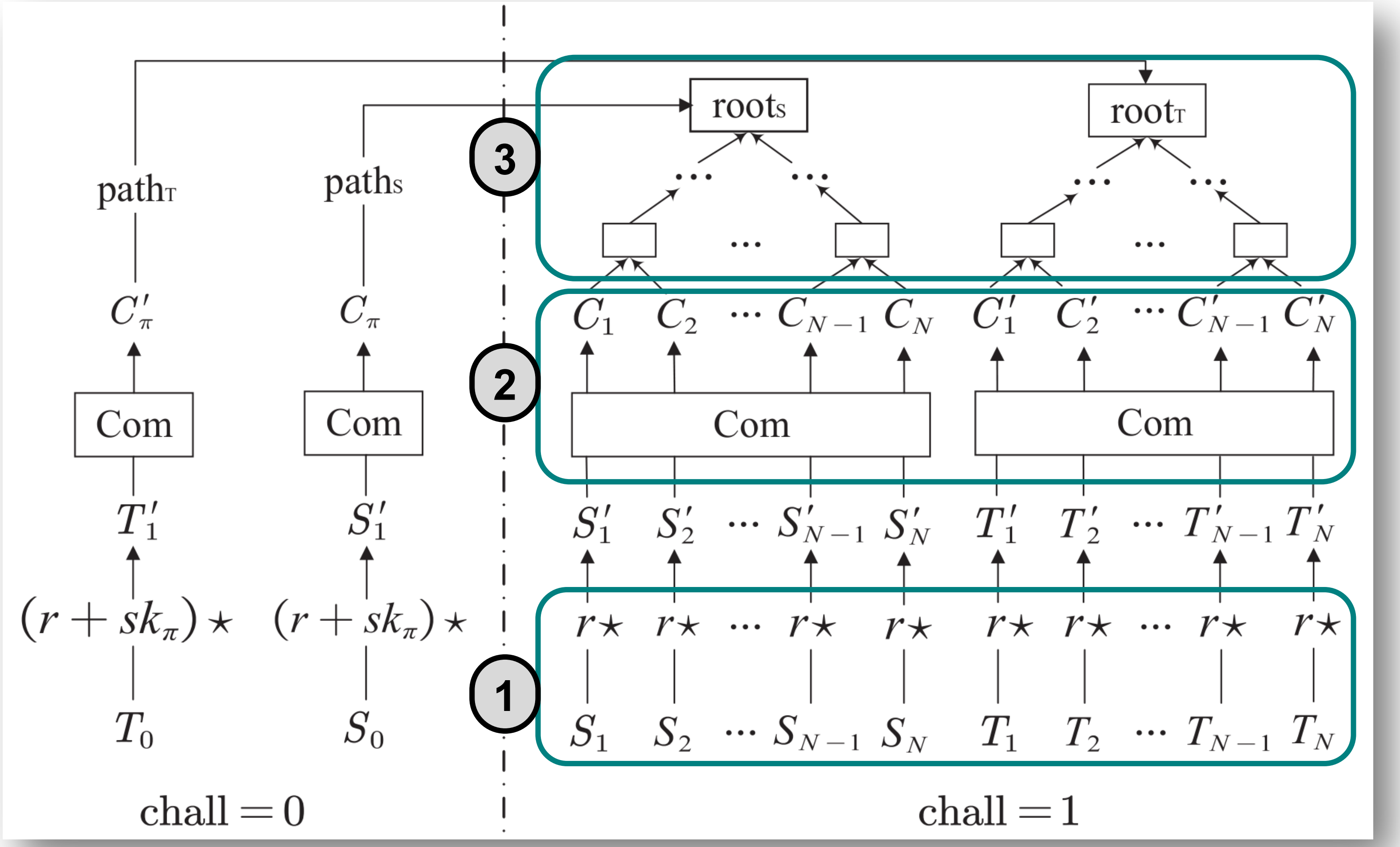


- **Commitment**
  1. Randomize rpk and TagSet
  2. Create commitments $C_i, C_i{}'$
  3. Create Merkle Tree
  4. Create the final commitment

$$com \leftarrow H_2(root_S, root_T)$$

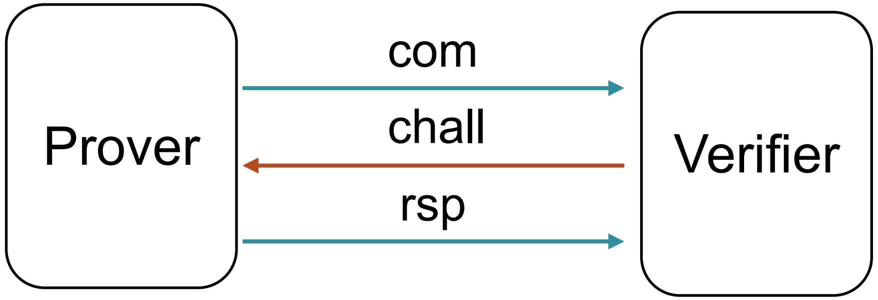
- **Challenge**

$$chall \leftarrow \{0,1\}$$



$$sk_\pi \star S_0 = S_\pi \text{ and } sk_\pi \star T_0 = T_\pi$$

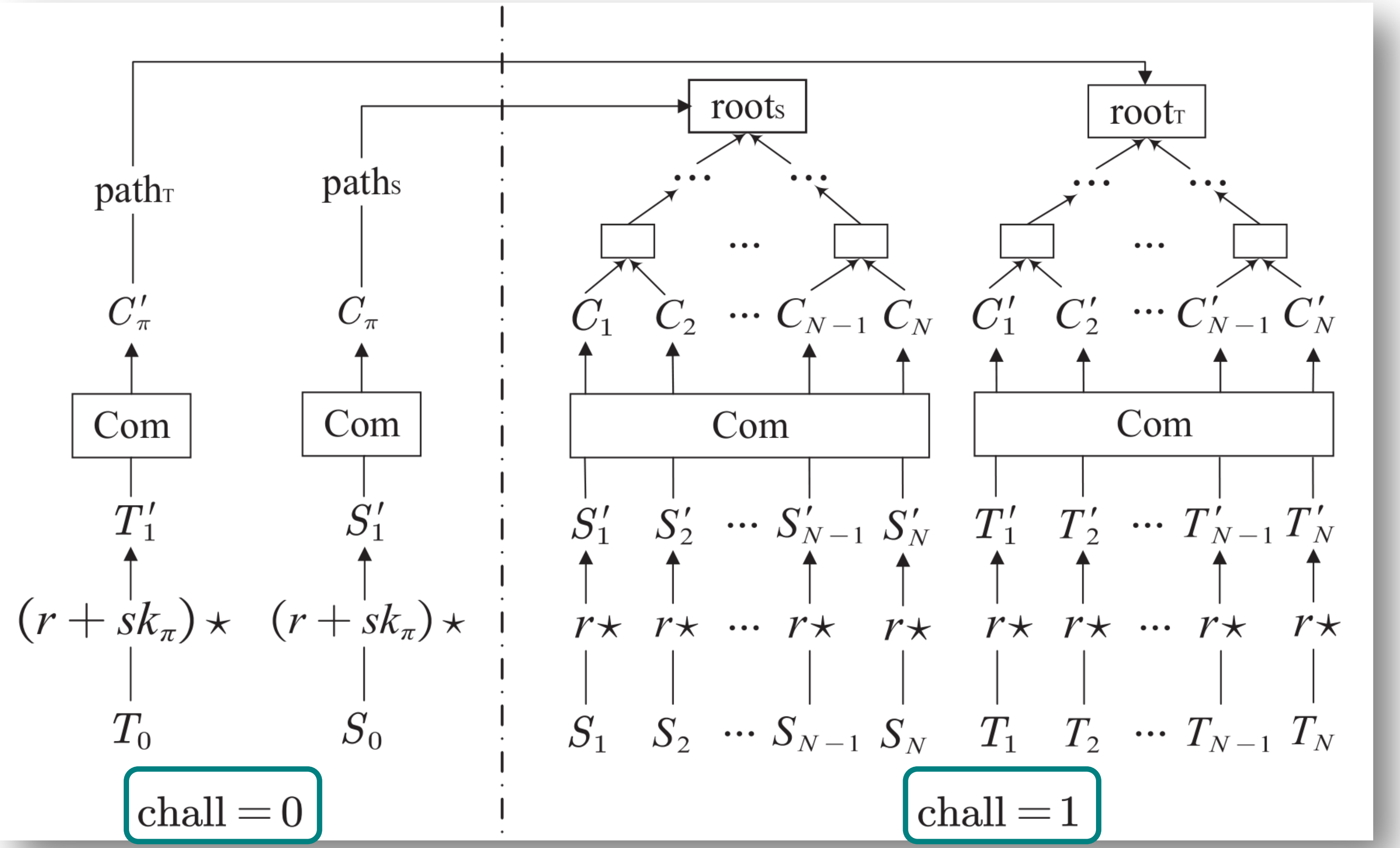
Construction — OR sigma protocol



- **Response**

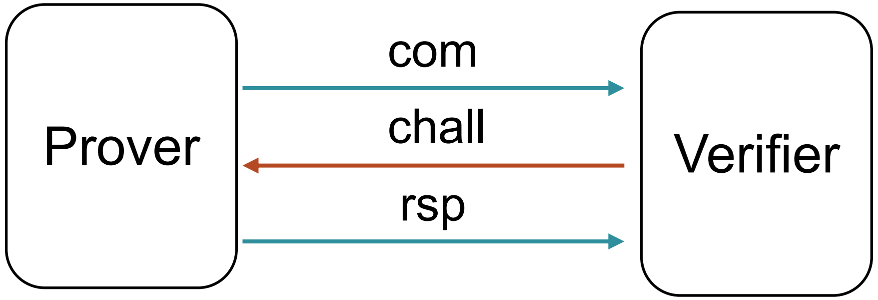
  If chall=0: The commitments C_π and $C_{\pi'}$ will be revealed.

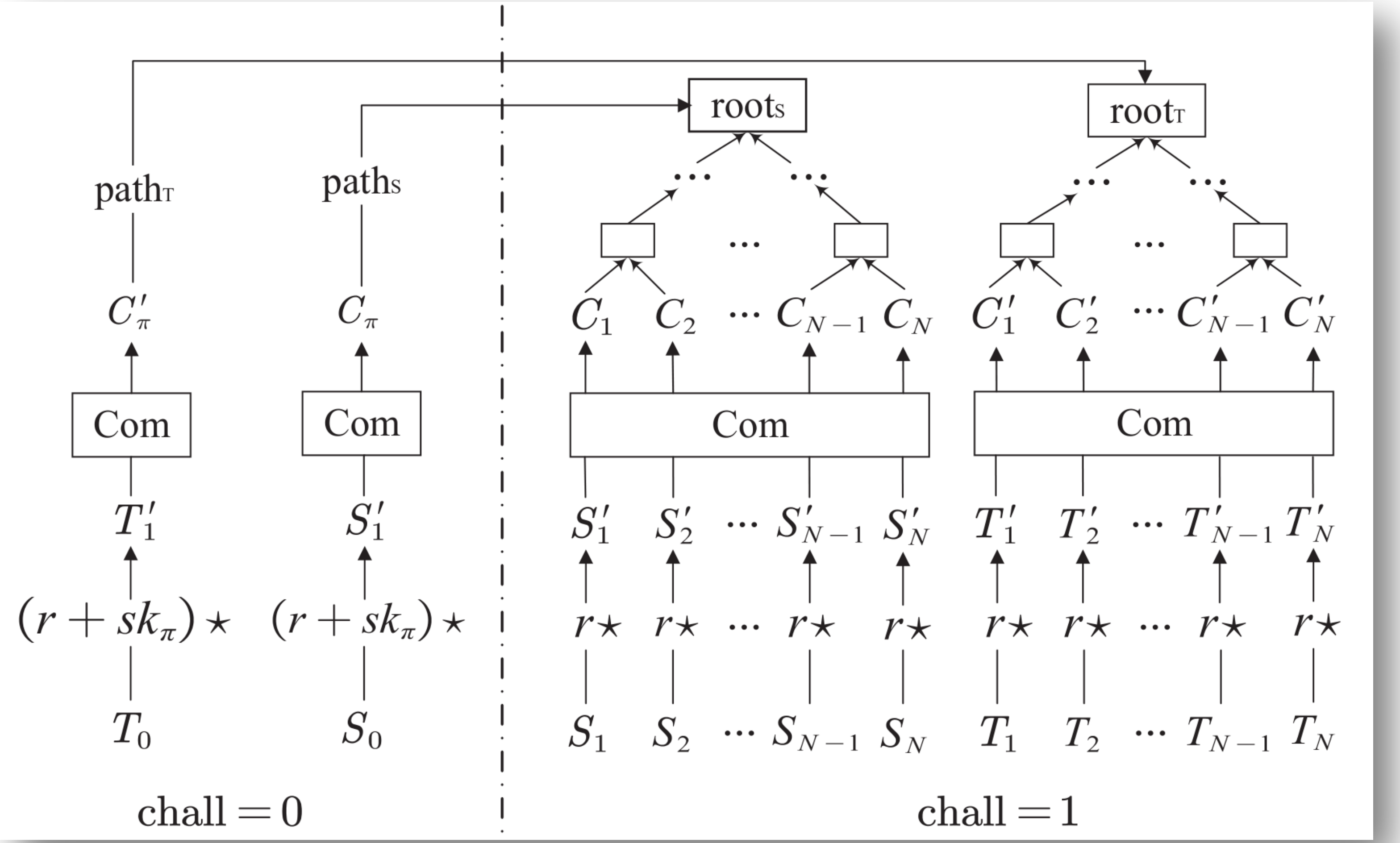
  If chall=1: All commitments will be revealed.



$$sk_\pi \star S_0 = S_\pi \text{ and } sk_\pi \star T_0 = T_\pi$$

Construction — OR sigma protocol





- **Verification**

  If chall=0

  1. Recovery root for rpk and TagSet from $\mathrm{path_T}$ and $\mathrm{path_S}$
  2. Verify the final commitment

  If chall=1

  1. Recovery root from all commitments
  2. Verify the final commitment

$$sk_\pi \star S_0 = S_\pi \text{ and } sk_\pi \star T_0 = T_\pi$$

Construction — Isogeny-based TRS scheme

Generate Tag Set

$\underline{\text{RSign\_ISO}((sk_\pi, \pi), L, M)}$

$(issue, \text{rpk}) \leftarrow L$
$T_0 = \mathcal{H}_1(L) \star S_0,\ a = \mathcal{H}_1(L, M)$
$T = (sk_\pi - \mathcal{H}_1(a, \pi)) \star T_0$
for all $i \in N$
  $k = \mathcal{H}_1(a, i)$
  $T_i = k \star T$
$\text{TagSet} \leftarrow (T_0, T_1, \ldots T_N)$
$\text{com} \leftarrow P^1_{main}(M, \text{rpk}, \text{TagSet})$
$\text{chall} \leftarrow \mathcal{H}_3(M, \text{rpk}, \text{TagSet}, \text{com})$
$\text{rsp} \leftarrow P^2_{main}((sk_\pi, \pi), \text{chall})$
return $\sigma = (T, \text{com}, \text{chall}, \text{rsp})$.

Add the tagSet to the OR-proof

$\underline{\text{RVer\_ISO}(L, M, \sigma)}$

$(issue, \text{rpk}) \leftarrow L$
$(T, \text{com}, \text{chall}, \text{rsp}) \leftarrow \sigma$
$T_0 = \mathcal{H}_1(L) \star S_0,\ a = \mathcal{H}_1(L, M)$
for all $i \in N$
  $k = \mathcal{H}_1(a, i)$
  $T_i = k \star T$
$\text{TagSet} \leftarrow (T_0, T_1, \ldots T_N)$
if $V^2_{main}(\text{com}, \text{chall}, \text{rsp}) = \text{accept}$
$\wedge \mathcal{H}_3(M, \text{rpk}, \text{TagSet}, \text{com}) = \text{chall}$
  return accept.
else return reject.

Recover two Tag Sets

$\underline{\text{RTrace\_ISO}(L, M, \sigma, M', \sigma')}$

$(issue, \text{rpk}) \leftarrow L$
$(T, \text{com}, \text{chall}, \text{rsp}) \leftarrow \sigma$
$(T', \text{com}', \text{chall}', \text{rsp}') \leftarrow \sigma'$
$a = \mathcal{H}_1(L, M), a' = \mathcal{H}_1(L, M')$
for all $i \in N$
  $k = \mathcal{H}_1(a, i), k' = \mathcal{H}_1(a', i)$
  $T_i = k \star T, T_i' = k' \star T'$
if for all $i \in [N]$, $T_i = T_i'$
  return linked.
if only exist one $i \in [N]$, such that $T_i = T_i'$
  return pk_i.
else return indep.

Link /Trace two signatures

$$H_1(a, \pi) \star \left((\text{sk}_\pi - H_1(a, \pi)) \star T_0 \right) = \text{sk}_\pi \star T_0 = T_\pi$$

Tracing the ring member π

Construction — Lattice-based TRS scheme

Generate Tag Set

Recover two Tag Sets

$\underline{\mathsf{RSign\_LAT}((sk_\pi, \pi), L, M)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $T_0 = \mathcal{H}_4(L),\ a = \mathcal{H}_5(L, M)$
3. $T_\pi = sk_\pi \star T_0$, $\mathsf{aux} = \frac{(T_\pi - a)}{\pi}$
4. **for** all $i \in N, i \neq \pi$
5. $\quad k = a + \mathsf{aux} \cdot i$
6. $\quad T_i = k \star T_0$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, \ldots T_N)$
8. $\mathsf{com} \leftarrow P^1_{main}(M, \mathsf{rpk}, \mathsf{TagSet})$
9. $\mathsf{chall} \leftarrow \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com})$
10. $\mathsf{rsp} \leftarrow P^2_{main}((sk_\pi, \pi), \mathsf{chall})$
11. **return** $\sigma = (\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp})$.

$\underline{\mathsf{RVer\_LAT}(L, M, \sigma)}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $T_0 = \mathcal{H}_4(L) \star S_0,\ a = \mathcal{H}_5(L, M)$
4. **for** all $i \in N$
5. $\quad k = a + \mathsf{aux} \cdot i$
6. $\quad T_i = k \star T_0$
7. $\mathsf{TagSet} \leftarrow (T_0, T_1, \ldots T_N)$
8. **if** $V^2_{main}(\mathsf{com}, \mathsf{chall}, \mathsf{rsp}) = \mathsf{accept}$ $\wedge \mathcal{H}_3(M, \mathsf{rpk}, \mathsf{TagSet}, \mathsf{com}) = \mathsf{chall}$
9. $\quad$ **return** accept.
10. **else return** reject.

$\underline{\mathsf{RTrace\_LAT}(L, M, \sigma, M', \sigma')}$

1. $(issue, \mathsf{rpk}) \leftarrow L$
2. $(\mathsf{aux}, \mathsf{com}, \mathsf{chall}, \mathsf{rsp}) \leftarrow \sigma$
3. $(\mathsf{aux}', \mathsf{com}', \mathsf{chall}', \mathsf{rsp}') \leftarrow \sigma'$
4. $a = \mathcal{H}_5(L, M), a' = \mathcal{H}_5(L, M')$
5. **for** all $i \in N$
6. $\quad k_i = a + \mathsf{aux} \cdot i$
7. $\quad k'_i = a' + \mathsf{aux}' \cdot i$
8. **if** for all $i \in [N]$, $k_i = k'_i$
9. $\quad$ **return** linked.
10. **if** only exist one $i \in [N]$, such that $k_i = k'_i$
11. $\quad$ **return** pk_i.
12. **else return** indep.

Add the tagSet to the OR-proof

Link /Trace two signatures

$\left(a + \frac{T_\pi - a}{\pi} \cdot \pi\right) = T_\pi$ **Tracing the ring member π**

Analysis — Correctness

- **completeness**

completeness can be deduced from the correctness of the main OR sigma protocol.

The traceability of the scheme in all possible situations.

- Situation 1 $(\pi = \pi' \wedge M = M')$ Linked
- Situation 2 $(\pi = \pi' \wedge M \neq M')$ pk_π
- Situation 3 $(\pi \neq \pi')$ Indep

- **security**

If the OR sigma protocol is soundness and zero-knowledge, the hash function H_1, H_2 are collision-resistant, the ResPGA is a restricted pair of group actions, then our TRS scheme satisfies **tag-linkability**, **anonymity** and **exculpability**.

Analysis — Performance

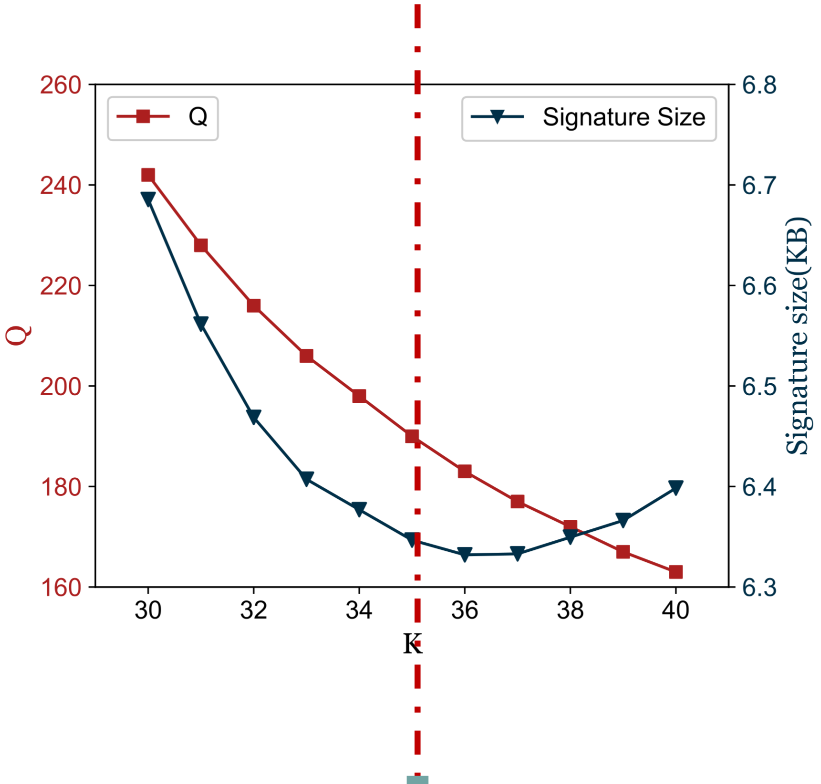
N			2^1	2^2	2^3	2^4	2^5	2^6
TRS_ISO	**Time**	KeyGen(ms)	39	39	39	39	39	39
		Sign(s)	3.37×10^1	6.63×10^1	1.31×10^2	2.64×10^2	5.23×10^2	1.07×10^3
		Verify(s)	3.20×10^1	6.02×10^1	1.16×10^2	2.31×10^2	4.64×10^2	9.22×10^2
	Size	Public Key(Byte)	64					
		Secret Key(Byte)	16					
		Signature(KB)	4.45	6.43	8.25	10.09	12.06	13.87
TRS_LAT (NIST 2)	**Time**	KeyGen(ms)	0.2	0.2	0.2	0.2	0.2	0.2
		Sign(ms)	68.5	101.3	131.4	230.8	390.3	764.0
		Verify(ms)	27.4	34.9	50.3	81.1	144.0	265.4
	Size	Public Key(Byte)	4096					
		Secret Key(Byte)	16					
		Signature(KB)	56.37	57.37	58.37	59.37	60.37	61.37

Analysis — Performance

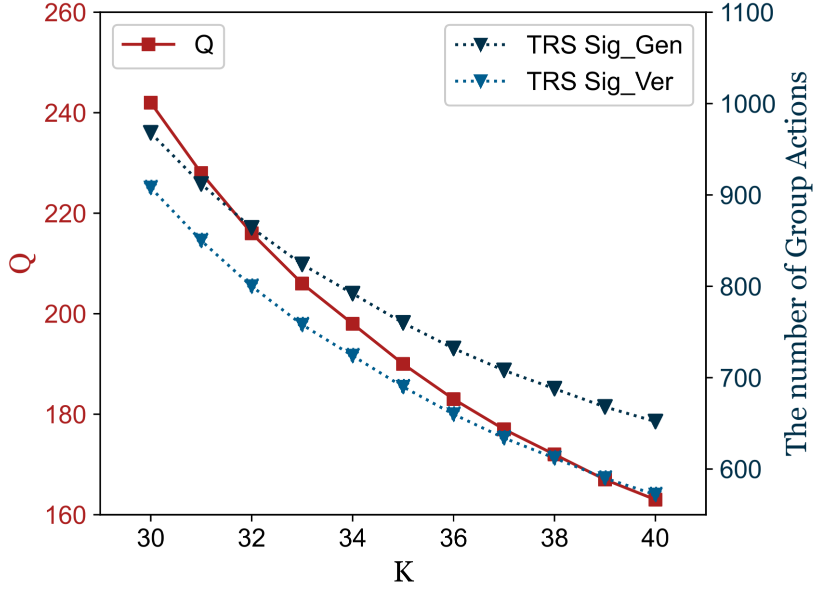
Schemes		Public key (KB)	Secret key (KB)	Signature size (KB)				Security Level
				2^1	2^3	2^6	2^{10}	
Calamari[4]		64 (Byte)	16 (Byte)	3.5	5.4	8.2	10	*
Beullens_ISO[3]		64 (Byte)	16 (Byte)	3.6	-	6.6	9.0	*
Raptor[27]		0.9	9.1	2.6	11	82	1331.2	100bits
Beullens_LAT[3]		5120 (Byte)	16 (Byte)	124	-	126	129	NIST 2
Falafl[4]		5120 (Byte)	16 (Byte)	49	50	52	55	NIST 2
Branco[6]		1577	0.5	-	1920	1536	245(MB)	NIST 5
Alessandra[32]		6	4	4	16	131	1024	NIST 5
Feng H[19]		-	-	135.1	136.3	138.2	140.7	NIST 5
Esign[17]		≤ 8.33	≤ 0.83	-	-	774	1021	NIST 5
this work	**ISO**	**64 (Byte)**	**16 (Byte)**	**4.5**	**8.3**	**13.9**	**22.2**	*
	LAT	**4096 (Byte)**	**16 (Byte)**	**56.3**	**58.3**	**61.3**	**65.3**	**NIST 2**
	LAT	**6144 (Byte)**	**16 (Byte)**	**74.3**	**76.3**	**79.3**	**83.3**	**NIST 5**

Analysis — Flexible

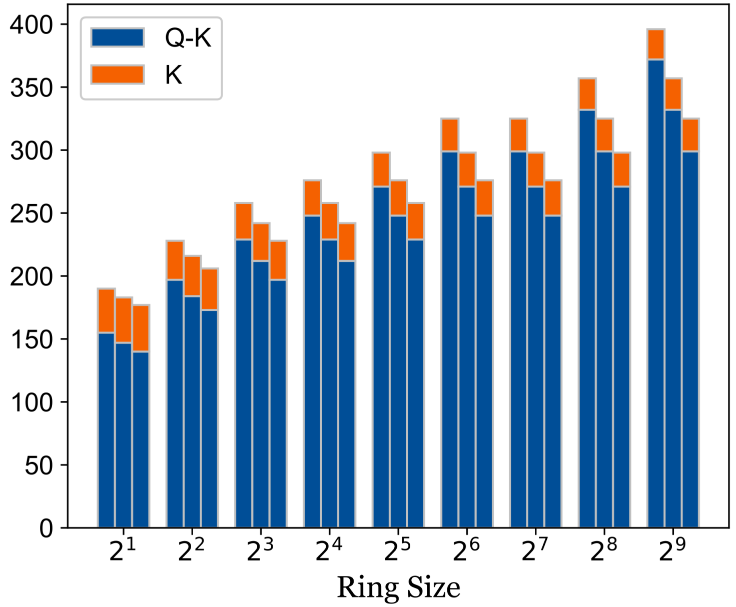
When $\binom{Q}{K} > 128$, the TRS scheme offers flexible customization of signature size and time for signature generation and verification.



the **minimum signature size** is obtained when **K = 36**



The **smaller** the value of Q is, the **less time** it takes for signature **generation and verification**.



three optimal (Q,K) pairs under different ring sizes

Conclusion

- A general traceable ring signature scheme is constructed.
- The first traceable ring signature scheme from isogeny is implemented.
- The signature size is logarithmic, the signature size and signing time are flexible.
- Futher topic:

  Reducing the number of group actions to minimize computational costs and extending the technique to other signature schemes.

Thanks!

Q&A