

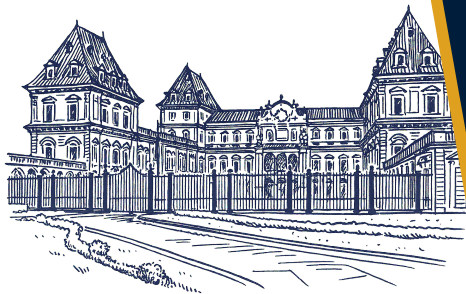


Politecnico  
di Torino

# Linkable Ring Signatures from Group Actions

Constructions, Pitfalls, and New Approaches  
Leonardo Errati

Chalmers S&P seminars, 29 Oct 2026



# Table of Contents

- Linkable ring signatures
- Group actions in cryptography
- LRS from group actions: BKP framework
- LRS from group actions: Canonical Lifting framework

# So many signatures!



| Primitive     | What is established?                                         | Motivation                                   |
|---------------|--------------------------------------------------------------|----------------------------------------------|
| Ordinary      | This key authorised $m$ .                                    | Software updates, authentication...          |
| Ring          | One key in a public set authorised $m$ .                     | Anonymous endorsement,<br>whistleblowing...  |
| Linkable Ring | As above, plus detection of repeated<br>use of the same key. | Duplicate voting, double spending...         |
| Group         | Membership in a managed group;<br>often authorised opening.  | Anonymous membership with<br>accountability. |

An organisation publishes a list of eligible public keys.



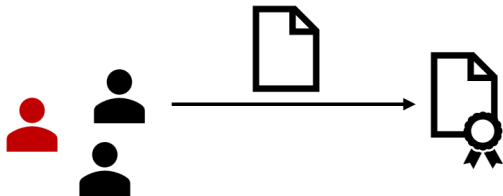
What we like:

- to learn that an eligible key signed.

What we don't:

- to learn which key signed.
- attackers able to forge a signature.

One signer and many public keys.



$$(pk_i, sk_i) \leftarrow \text{KeyGen}(pp)$$

$$\mathcal{R} = \{pk_1, \dots, pk_N\}$$

$$\sigma \leftarrow \text{Sign}(sk_i, m, \mathcal{R})$$

$$b \leftarrow \text{Verify}(\mathcal{R}, m, \sigma), \quad b \in \{0, 1\}$$

# Ring signatures: trouble in the application layer



Yes,

Users generate their own keys.  
Effective signer selects public keys.  
Verification binds signature to message and ring.

but...

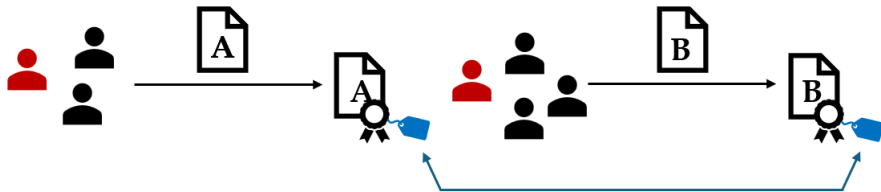
Who is eligible to register keys?  
Is the ring fixed, are decoys sampled, etc?  
How are keys authenticated, validated, revoked?

## More troubles

A large ring is not automatically a large anonymity set (malicious keys, metadata, overlapping rings...).



Publicly recognise repeated use of a signing key without identifying it. Often implemented with a **linking tag** bound to the signing key.



$$\sigma = (\sigma', \tau)$$

$$b \leftarrow \text{Link}(\sigma_a, \sigma_b), \quad b \in \{0, 1\}.$$

A controlled loss of privacy able to reveal repetition.

Liu, Wei, and Wong, "Linkable spontaneous anonymous group signature for ad hoc groups", *Australasian conference on information security and privacy*, 2004

| Application      | Why anonymity?                 | RS or LRS?                                                 |
|------------------|--------------------------------|------------------------------------------------------------|
| Private payments | Hide which output is spent.    | LRS: reject repeated spending.                             |
| Voting / survey  | Hide the participating member. | LRS: one use per key and context.                          |
| Whistleblowing   | Protect an insider's identity. | RS for unlinkability; LRS to recognise a returning source. |

Key policy must be part of the application. Decide eligibility, ring selection, link scope, revocation...

## Example: Monero's key images



A payment hides which output it spent among a ring of real plus decoy outputs.



Double spending repeats  $I$ , even with a different ring.

### Key image $I$ ?

A public fingerprint derived from the spent output's secret key. Repeats on reuse.

## Example: anonymous voting

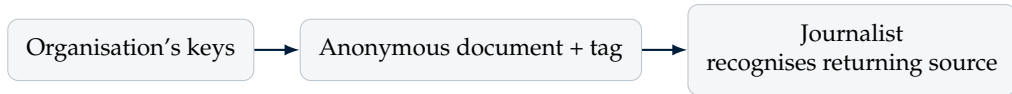


LRS does not provide ballot secrecy, coercion resistance, one-person-one-key registration, verifiable tally...

### Which scope?

Persistent keys with global linking? Dedicated per-election keys?

# Example: whistleblowing



## Crypto-Dissent, anonymous chat

Authenticate members before joining a chat.  
Recognise repeat authentication by the same credential.

## Crypto-Drop, anonymous sources

Authenticate a document as coming from an organisation's member. Journalists can link later submissions from the same source.

## Metadata

LRS hide the signing key within the ring, not IP addresses or document metadata.

Idea: with  $N$  keys an attacker cannot create  $N + 1$  valid and pairwise unlinked signatures.

## Game $_{\mathcal{A}}^{\text{Link}}(\lambda)$

```
1:  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ 
2:  $(PK, \{(\sigma_j, m_j, \mathcal{R}_j)\}_{j=1}^{N+1}) \leftarrow \mathcal{A}(\text{pp})$ , where  $|PK| = N$ 
3: for  $j = 1, \dots, N + 1$  do
4:   if  $\mathcal{R}_j \not\subseteq PK$  or  $\text{Verify}(\mathcal{R}_j, m_j, \sigma_j) \neq \text{accept}$  then
5:     return 0
6:   end if
7: end for
8: for all  $i < j$  do
9:   if  $\text{Link}(\sigma_i, \sigma_j) \neq \text{reject}$  then return 0
10: end for
11: return 1
```

Idea: link pattern must be visible but identities must be hidden.

## $\text{Game}_{\mathcal{A}}^{\text{L-Anon}}(\lambda)$

```
1:  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ 
2:  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}), i \in [N]$ 
3:  $PK \leftarrow \{\text{pk}_i : i \in [N]\}; b \xleftarrow{\$} \{0, 1\}$ 
4:  $(\text{pk}_0^*, \text{pk}_1^*) \leftarrow \mathcal{A}(\text{pp}, PK)$ 
5: Require distinct  $\text{pk}_0^*, \text{pk}_1^* \in PK$ 
6:  $K \leftarrow \{(\text{pk}_i, \text{sk}_i) : \text{pk}_i \notin \{\text{pk}_0^*, \text{pk}_1^*\}\}$ 
7:  $b^* \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}, b}}(K)$ 
8: return  $[b^* = b]$ 
```

## $\mathcal{O}_{\text{sign}, b}(\text{pk}, m, \mathcal{R})$

```
1: Require  $\text{pk} \in \{\text{pk}_0^*, \text{pk}_1^*\} \subseteq \mathcal{R}$ 
2: if  $\text{pk} = \text{pk}_0^*$  then
3:   return  $\text{Sign}(\text{sk}_b^*, m, \mathcal{R})$ 
4: else
5:   return  $\text{Sign}(\text{sk}_{1-b}^*, m, \mathcal{R})$ 
6: end if
```

For input  $\text{pk}_j^*$ , channel  $j$  signs with  $\text{sk}_{j \oplus b}^*$ .  
Non-challenge keys in  $\mathcal{R}$  may be adversarial.

Idea: link pattern must be visible but identities must be hidden **under one signature**.

## $\text{Game}_{\mathcal{A}}^{1\text{-Anon}}(\lambda)$

- 1:  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$
- 2:  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}), i \in [N]$
- 3:  $PK \leftarrow \{\text{pk}_i : i \in [N]\}; b \xleftarrow{\$} \{0, 1\}$
- 4:  $(\text{pk}_0^*, \text{pk}_1^*) \leftarrow \mathcal{A}(\text{pp}, PK)$
- 5: Require distinct  $\text{pk}_0^*, \text{pk}_1^* \in PK$
- 6:  $K \leftarrow \{(\text{pk}_i, \text{sk}_i) : \text{pk}_i \notin \{\text{pk}_0^*, \text{pk}_1^*\}\}$
- 7:  $\text{used} \leftarrow 0$
- 8:  $b^* \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}, b}^{(1)}}(K)$
- 9: **return**  $[b^* = b]$

## $\mathcal{O}_{\text{sign}, b}^{(1)}(\text{pk}, m, \mathcal{R})$

- 1: Require  $\text{pk} \in \{\text{pk}_0^*, \text{pk}_1^*\} \subseteq \mathcal{R}$
- 2: **if**  $\text{used} = 1$  **then return**  $\perp$
- 3:  $\text{used} \leftarrow 1$
- 4: **if**  $\text{pk} = \text{pk}_0^*$  **then**
- 5:     **return**  $\text{Sign}(\text{sk}_b^*, m, \mathcal{R})$
- 6: **else**
- 7:     **return**  $\text{Sign}(\text{sk}_{1-b}^*, m, \mathcal{R})$
- 8: **end if**

Beullens, Katsumata, and Pintore, "Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices", *International Conference on the Theory and Application of Cryptology and Information Security*, 2020

Bultel and Olivier-Anclin, "On the Anonymity of Linkable Ring Signatures", *CANS 2024*, 2024

An attacker cannot create a fresh valid signature that links to an honest user's signature.

## Game $_{\mathcal{A}}^{\text{Non-Frame}}(\lambda)$

- 1:  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$
- 2:  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}), i \in [N]$
- 3:  $PK \leftarrow \{\text{pk}_i : i \in [N]\}$
- 4:  $SL, CL, T \leftarrow \emptyset$
- 5:  $(\mathcal{R}^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}, \mathcal{O}_{\text{corrupt}}}(\text{pp}, PK)$
- 6: **return** the conjunction:  
     $\text{Verify}(\mathcal{R}^*, m^*, \sigma^*) = \text{accept}$   
     $(\cdot, m^*, \mathcal{R}^*) \notin SL$   
     $\exists (i, \sigma) \in T : \text{pk}_i \notin CL \wedge \text{Link}(\sigma^*, \sigma) = \text{accept}$

$SL$ : signing queries  $(i, m, \mathcal{R})$ .

$T$ : returned signatures  $(i, \sigma)$ .     $CL$ : corrupted public keys.

Beullens, Katsumata, and Pintore, "Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices", *International Conference on the Theory and Application of Cryptology and Information Security*, 2020

## $\mathcal{O}_{\text{sign}}(i, m, \mathcal{R})$

- 1: Require  $i \in [N]$  and  $\text{pk}_i \in \mathcal{R}$
- 2:  $\sigma \leftarrow \text{Sign}(\text{sk}_i, m, \mathcal{R})$
- 3:  $SL \leftarrow SL \cup \{(i, m, \mathcal{R})\}$
- 4:  $T \leftarrow T \cup \{(i, \sigma)\}$
- 5: **return**  $\sigma$

## $\mathcal{O}_{\text{corrupt}}(i)$

- 1: Require  $i \in [N]$
- 2:  $CL \leftarrow CL \cup \{\text{pk}_i\}$
- 3: **return**  $(\text{pk}_i, \text{sk}_i)$

An attacker cannot sign for a ring consisting entirely of uncorrupted users.

## Game $_{\mathcal{A}}^{\text{Unf}}(\lambda)$

- 1:  $\text{pp} \leftarrow \text{Setup}(1^\lambda)$
- 2:  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\text{pp}), i \in [N]$
- 3:  $PK \leftarrow \{\text{pk}_i : i \in [N]\}; SL, CL \leftarrow \emptyset$
- 4:  $(\mathcal{R}^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}, \mathcal{O}_{\text{corrupt}}}(\text{pp}, PK)$
- 5: **return** the conjunction:
  - $\mathcal{R}^* \subseteq PK \setminus CL$
  - $(\cdot, m^*, \mathcal{R}^*) \notin SL$
  - $\text{Verify}(\mathcal{R}^*, m^*, \sigma^*) = \text{accept}$

(oracles and logs as before)

## Implied by the other properties

Compare a forgery with one fresh honest signature per ring key.

- Links to an honest signature?  
**Non-frameability** is broken.
- No such link?  
**Linkability** is broken.

1. Only one member actively signs.
2. In LRS, we also reveal repeated use.
3. So many application-layer decisions! Eligibility, ring policy, and so on.

# Table of Contents

- Linkable ring signatures
- Group actions in cryptography
- LRS from group actions: BKP framework
- LRS from group actions: Canonical Lifting framework

# Why post-quantum?



## **Assumption diversity:**

quantum attacks (Shor, Grover)  
threaten past primitives.

## **Functional diversity:**

post-quantum ordinary  
signature does not  
automatically get advanced  
functionalities.



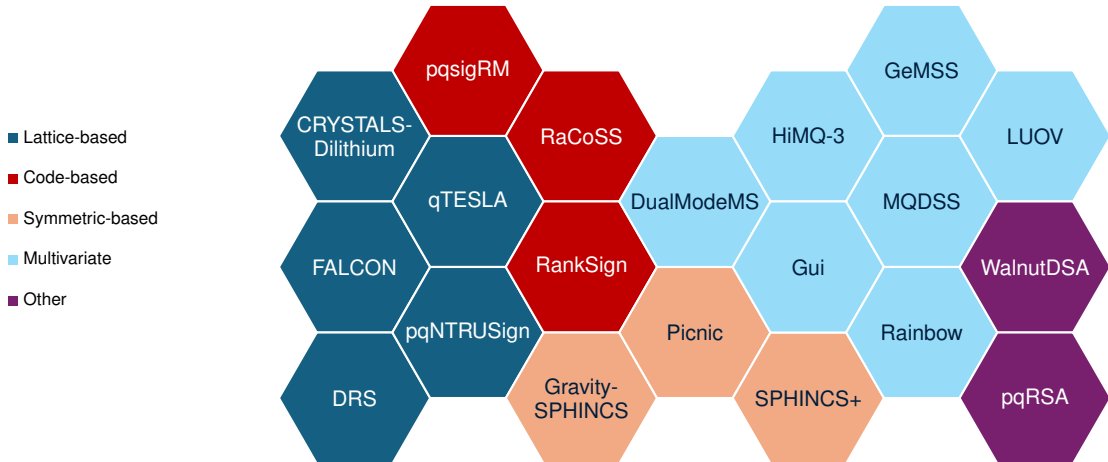
NIST IR 8547 (October 2026): transition plan for digital signatures.

| Digital Signature Algorithm Family | Parameters                                     | Transition                                                                 |
|------------------------------------|------------------------------------------------|----------------------------------------------------------------------------|
| <b>ECDSA</b><br>[FIPS186]          | $\geq 112$ bits of classical security strength | <b><i>Deprecated</i></b> after 2030<br><b><i>Disallowed</i></b> after 2035 |
| <b>EdDSA</b><br>[FIPS186]          | $\geq 128$ bits of classical security strength | <b><i>Deprecated</i></b> after 2030<br><b><i>Disallowed</i></b> after 2035 |
| <b>RSA</b><br>[FIPS186]            | $\geq 112$ bits of classical security strength | <b><i>Deprecated</i></b> after 2030<br><b><i>Disallowed</i></b> after 2035 |

Moody, Perlner, Regenscheid, Robinson, and Cooper, *Transition to Post-Quantum Cryptography Standards*, National Institute of Standards and Technology, NIST Internal Report 8547, 2026, Table 2 and Secs. 4.1-4.1.1, pp. 14-15

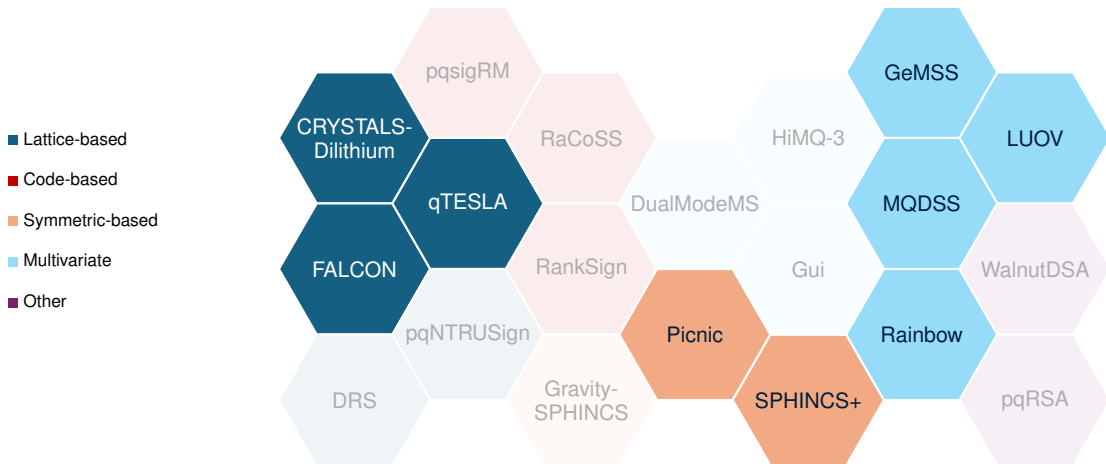
# Original NIST call (signatures)

Round 1, 20 December 2017: 20 candidates



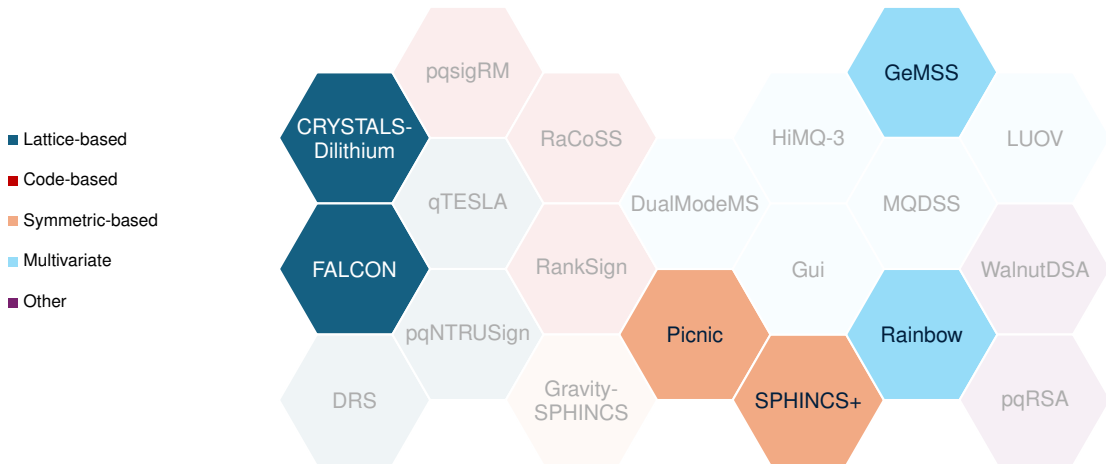
# Original NIST call (signatures)

Round 2, 30 January 2019: 9 candidates



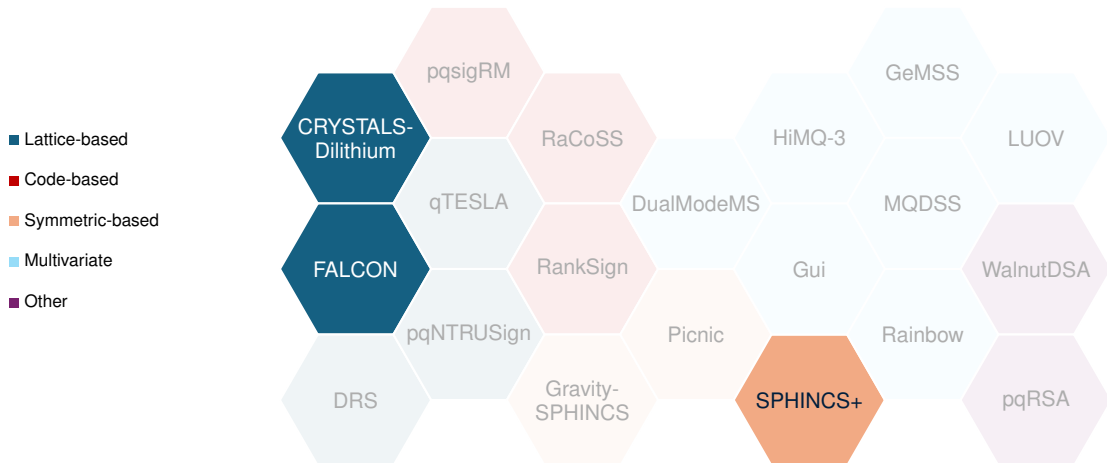
# Original NIST call (signatures)

Round 3, 22 July 2020: 3 finalists and 3 alternates



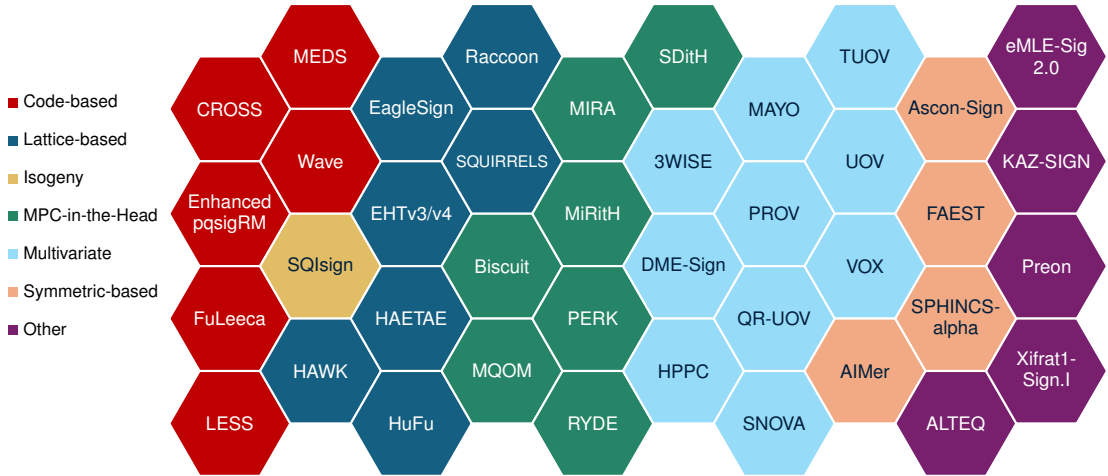
# Original NIST call (signatures)

5 July 2022: 3 selected



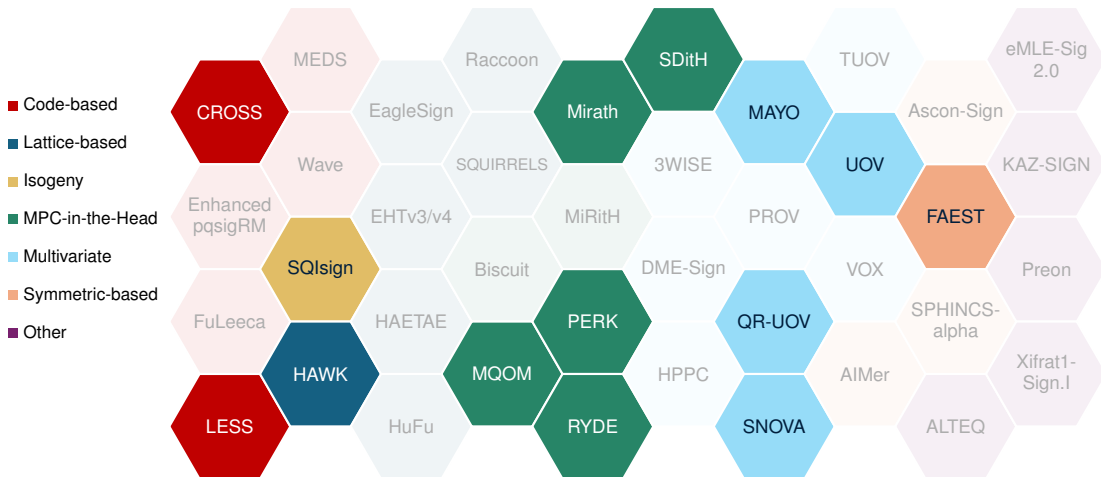
# NIST Call for additional signatures

Round 1, 17 July 2023: 40 candidates



# NIST Call for additional signatures

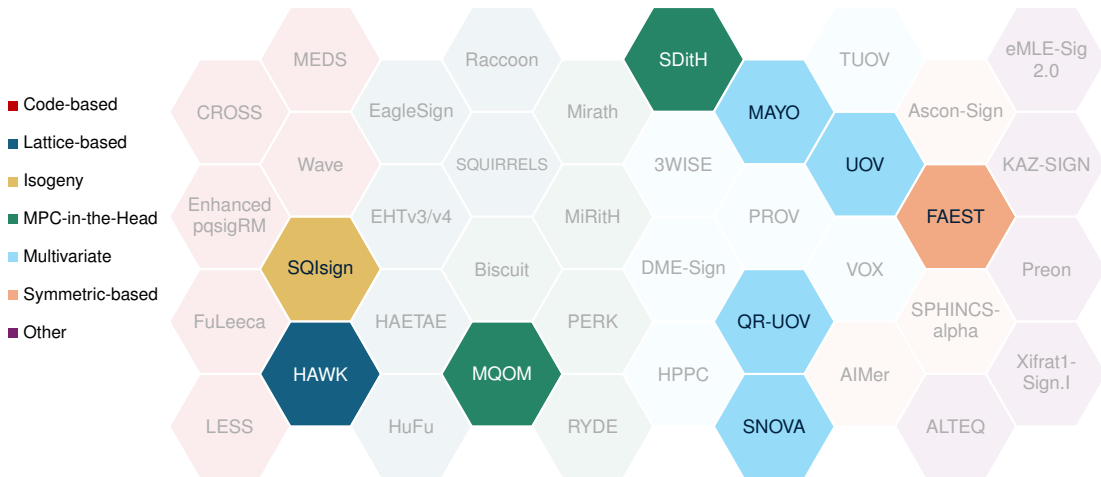
Round 2, 24 October 2024: 14 candidates



MIRA and MiRitH merged in Mirath.

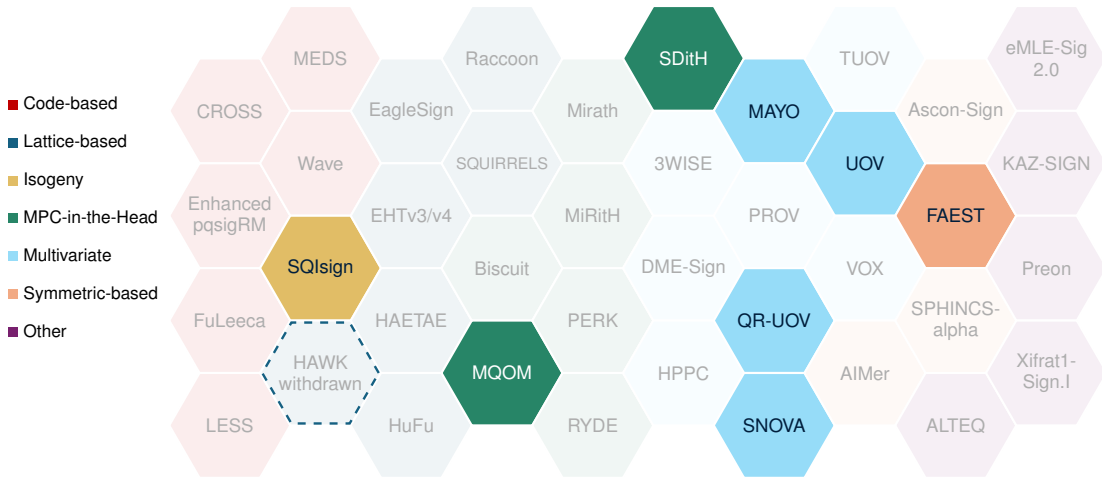
# NIST Call for additional signatures

Round 3, 14 May 2026: 9 candidates



# NIST Call for additional signatures

October 2026: 8 candidates



HAWK withdrawn, 29 July 2026.



$(G, \cdot)$ : transformations, with composition.  
 $X$ : objects on which the transformations act.

Definition: a left group action

A map  $\star : G \times X \rightarrow X$  such that

$$e \star x = x, \quad (gh) \star x = g \star (h \star x).$$

Commutativity is not required. For Rubik's cube,

$X = \{\text{possible states of the cube}\},$

$$G \cong (\mathbb{Z}_3^7 \times \mathbb{Z}_2^{11}) \rtimes ((A_8 \times A_{12}) \rtimes \mathbb{Z}_2) \leq S_{48}.$$

Recovering a secret transformation from input and output should be hard.

$$\underbrace{x}_{\text{public base}} \xrightarrow{g} \underbrace{y = g \star x}_{\text{public key}}.$$

## Group Action Inverse Problem

Given  $(x, y)$ , find some  $g$  with  $y = g \star x$ . The witness need not be unique.

Some instantiations are post-quantum!

| Objects      | Transformations                      | Hard problem        |
|--------------|--------------------------------------|---------------------|
| Linear codes | Coordinate permutations and scalings | Code equivalence    |
| Lattices     | Integral changes of basis            | Lattice isomorphism |

# Example: code equivalence



**Objects:**  $k$ -dimensional linear codes  $C = \text{rowspan}(G) \subseteq \mathbb{F}_q^n$ .

**Transformations:** coordinate permutations  $P$  and non-zero scalings  $D$ .

$$G' = SGPD, \quad S \in \text{GL}_k(\mathbb{F}_q), \quad C' = C \cdot (PD).$$

$$\begin{array}{ccc} G & & GPD & & G' = SGPD \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \xrightarrow{PD} & \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix} & \xrightarrow{S} & \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 0 \end{pmatrix} \end{array}$$

**Hard problem:** recover a coordinate transformation.

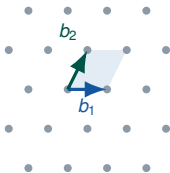
# Example: lattice isomorphism



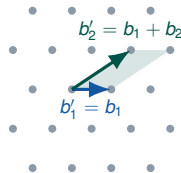
**Objects:** Gram matrices  $Q = B^\top B$  (basis lengths & angles).

**Transformations:** unimodular matrices  $U \in \text{GL}_n(\mathbb{Z})$ , i.e.  $\det U = \pm 1$ .

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B' = BU, \quad Q' = Q \cdot U = U^\top Q U \quad (\text{right action}).$$



$$B = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}, \quad Q = \begin{pmatrix} 4 & 2 \\ 2 & 5 \end{pmatrix}.$$



$$B' = \begin{pmatrix} 2 & 3 \\ 0 & 2 \end{pmatrix}, \quad Q' = \begin{pmatrix} 4 & 6 \\ 6 & 13 \end{pmatrix}.$$

**Hard problem:** given  $Q, Q'$ , find a suitable  $U$ .

1. A group describes transformations, the action describes their effect.
2. Must be efficient to compute but with hard computational problems.
3. Codes and lattices provide candidate hard problems.

# Table of Contents

- Linkable ring signatures
- Group actions in cryptography
- LRS from group actions: BKP framework
- LRS from group actions: Canonical Lifting framework

Prover knows  $g$  such that  $y = g \star x$ .

$$\begin{array}{ccc} x & \xrightarrow{g} & y = g \star x \\ & \searrow \rho g & \downarrow \rho \\ & & c = \rho \star y \end{array}$$

$$\mathcal{R}_{\text{GA}} = \{((x, y), g) : y = g \star x\}.$$

Challenge 0  
reveal  $\rho$ ; check  $c = \rho \star y$

Challenge 1  
reveal  $\rho g$ ; check  $c = (\rho g) \star x$

Fix the public base  $x$  and ring  $(y_1, \dots, y_N)$ .

$$\mathcal{R}_{\text{GA}}^{\text{OR}} = \{((x, y_1, \dots, y_N), (i, g)) : i \in [N] \wedge y_i = g \star x\}.$$

1. Randomise keys applying the same mask  $\rho$ , then commit with independent  $r_j$ :

$$a_j = \rho \star y_j, \quad c_j = \text{Com}(a_j; r_j), \quad j \in [N].$$

2. **Index-hiding Merkle tree:** leaves  $H(c_j)$ , sorted child hashes  $u, v$ :  $H(\min(u, v) \parallel \max(u, v))$ .

Challenge 0: reconstruct every branch  
Reveal the seed generating  $\rho$  and all  $r_j$ .  
Recompute all  $c_j$  from the  
public ring, check the root.

Challenge 1: open the known branch  
Reveal  $z = \rho g, r_i$ , and an index-hiding path.  
Reconstruct  $\text{Com}(z \star x; r_i)$ ,  
check its path against  $R$ .

Given two actions

$$\star : G \times X \rightarrow X, \quad \bullet : G \times T \rightarrow T.$$

prove that the same witness generates one public key and the tag:

$$\mathcal{R}_{\text{GA}}^{\text{OR-t}} = \{((x, y_1, \dots, y_N, t, \tau), (i, g)) : y_i = g \star x \wedge \tau = g \bullet t\}.$$

Parallel repetitions + Fiat-Shamir = LRS.

Link using a tag relation (equality, closeness...).

$$\underbrace{y = g \star x,}_{\text{public key}} \quad \underbrace{\tau = g \bullet t,}_{\text{linking tag}} \quad \underbrace{z = \rho g}_{\text{masked proof response}} .$$

| Desired guarantee  | What the pair of actions must prevent                          |
|--------------------|----------------------------------------------------------------|
| Linkability        | Two witnesses for one public key yielding unlinked tags.       |
| Linkable anonymity | Recognising which public key is correlated with a visible tag. |
| Non-frameability   | Finding a new witness whose tag links to an honest user's tag. |

$$\underbrace{y = g \star x}_{\text{public key}}, \quad \underbrace{\tau = g \bullet t}_{\text{linking tag}}, \quad \underbrace{z = \rho g}_{\text{masked proof response}}.$$

1. Efficient algorithms for the actions, sampling, evaluation, encoding.
2. Efficient algorithm for evaluating  $\text{Link}_{\text{GA}}(\tau, \tau')$ , with  $\text{Link}_{\text{GA}}(\tau, \tau) = 1$ .
3. Accepted responses lie in a common, efficiently recognisable domain and have a distribution independent of the secret  $g$ .

More related to security properties:

4. The public pair must look like independent uniform elements of  $X \times T$ .
5. Two valid witnesses for the same public key must not yield unlinked tags.
6. Finding a new witness whose tag links to it must be hard.

Two instantiations:

**Calamari**

**Isogenies (CSIDH).**

**Falaf**

**Module lattices (MLWE/MSIS).**

# Attack 1: Don't Use It Twice!

Reusing the **same secret transformation** can expose more information.

We have two views of the same hidden relabelling. One instance hides a permutation and coordinate scalings. A second instance adds constraints!

$$C_1 \xrightarrow{Q} C'_1 = C_1 Q$$

$$C_2 \xrightarrow{Q} C'_2 = C_2 Q$$

A candidate transformation must fit both pairs. The second pair can eliminate the ambiguity protecting the first.

# Attack 1: Don't Use It Twice! (ctd)



$$C'_1 = C_1 Q, \quad C'_2 = C_2 Q \implies G_j Q (H'_j)^\top = 0.$$

**Public inputs:**  $G_j$  generates  $C_j$ ; compute a parity-check matrix  $H'_j$  for  $C'_j$ , so  $G'_j (H'_j)^\top = 0$ .

|                          |
|--------------------------|
| $M_1 = H'_1 \otimes G_1$ |
| $M_2 = H'_2 \otimes G_2$ |

vec( $Q$ )

= 0

Stack both linear systems.  
The same secret  $Q$  must satisfy both.

Guess one nonzero position of the *monomial*  $Q$ , every other entry in that row and column must be zero. At rate  $k/n = 1/2$ , the paper gives a heuristic polynomial-time recovery attack, supported by experiments.

## Attack 2: Don't Use It Twice! Reloaded



For a public Gram matrix  $Q_0$ , the proposal uses a secret unimodular  $U$ :

$$Y = U^\top Q_0 U \text{ public key,} \quad T = U^{-\top} Q_0 U^{-1} \text{ tag.}$$

Rearranging the tag equation gives  $Q_0 = U^\top T U$ . The key and tag therefore provide **two public pairs with the same secret**:

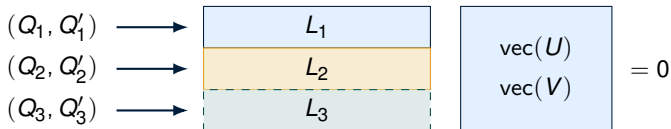
$$\begin{array}{ccc} Q_0 & \xrightarrow{U} & Y \\ T & \xrightarrow{\text{same } U} & Q_0 \end{array}$$

Testing whether a candidate key and the tag share this pattern can identify the signer.

## Attack 2: Don't Use It Twice! Reloaded

$$Q'_j = U^\top Q_j U \implies Q'_j V - U^\top Q_j = 0, \quad V = U^{-1}.$$

Temporarily treat  $U, V$  as independent unknowns.



### 2 instances: distinguish.

The rank detects enough information on the shared secret to break linkable anonymity.

### 3 instances: recover.

The kernel constrains  $(U, V)$ , enforce  $V = U^{-1}$  to recover  $\pm U$ .

Both results rely on heuristic rank assumptions, supported by experiments (Assumptions 1 and 2 in the paper).

## Attack 3: non-commutativity



### Correctness

One response  $Z$  must work for both

key check  $Z = UR$ ,

tag check  $Z = RU$ .

**An honest signature can be rejected.**

### Linkability

Different secret matrices can describe the same public key. The signer might get one public key with different tags. The same key can evade linking.

1. The same secret generates one public key in the ring and the linking tag. They must be secure together.
2. LRS security requires separate assumptions.

# Table of Contents

- Linkable ring signatures
- Group actions in cryptography
- LRS from group actions: BKP framework
- LRS from group actions: Canonical Lifting framework

For a group action  $(G, X, \star)$ , take efficient maps

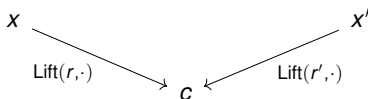
$$\text{Rep} : G \rightarrow R, \quad \text{Can} : X \rightarrow C, \quad \text{Lift} : R \times X \rightarrow C.$$

$$\begin{array}{ccc} G \times X & \xrightarrow{(\text{Rep}, \text{id})} & R \times X \\ \star \downarrow & & \downarrow \text{Lift} \\ X & \xrightarrow{\text{Can}} & C \end{array}$$

$$\text{Can}(g \star x) = \text{Lift}(\text{Rep}(g), x).$$

Given two public objects  $x, x' \in X$ , find representatives  $r, r' \in R$  such that

$$\text{Lift}(r, x) = \text{Lift}(r', x').$$



A canonical lifting is **effective** if we have efficient algorithms for

1. Given any  $r \in R$ , find  $g \in G$  such that  $\text{Rep}(g) = r$ .
2. Given  $u, v \in X$  with  $\text{Can}(u) = \text{Can}(v)$ , find  $h \in G$  such that  $v = h \star u$ , if any.

For an effective lifting, CLCP and GAIP are polynomial-time equivalent.

## Same full witness

$$y = g \star x, \quad \tau = g \bullet t.$$

## Just same representative

$$y = g \star x, \quad \tau = \tilde{g} \star t, \\ \text{Rep}(g) = \text{Rep}(\tilde{g}).$$

**Canonical lifting** lets the verifier check this shared representative.

Suppose  $y = g \star x$ ,  $\tau = \tilde{g} \star t$ , and  $\text{Rep}(g) = \text{Rep}(\tilde{g})$ . A canonical lifting is **left-compatible** if:

1. For all  $g, \tilde{g}, \rho \in G$ ,

$$\text{Rep}(g) = \text{Rep}(\tilde{g}) \implies \text{Rep}(\rho g) = \text{Rep}(\rho \tilde{g})$$

2. Given  $g \in G$ , we can efficiently sample uniformly from  $[g]_{\text{Rep}} = \{h \in G : \text{Rep}(h) = \text{Rep}(g)\}$ .

Then one response  $r = \text{Rep}(\rho g) = \text{Rep}(\rho \tilde{g})$  checks both:

$$\begin{array}{ccc} x & \xrightarrow{g} & y \\ & \searrow \rho g & \downarrow \rho \\ & & \rho \star y \end{array} \qquad \begin{array}{ccc} t & \xrightarrow{\tilde{g}} & \tau \\ & \searrow \rho \tilde{g} & \downarrow \rho \\ & & \rho \star \tau \end{array}$$

$$\text{Lift}(r, x) = \text{Can}(\rho \star y), \quad \text{Lift}(r, t) = \text{Can}(\rho \star \tau).$$

$$\text{Rep}(S, PD) = P$$

discard  $S$  and  $D$

$$\text{Can}(SAD) = \text{Can}(A)$$

Use the canonical form of Duong et al.

$$\text{Lift}(P, A) = \text{Can}(AP)$$

Permute the columns of  $A$  by  $P$ , then apply Can.

## Public key and linking tag

Fix two public generator matrices  $G_0, T_0 \in \mathbb{F}_q^{k \times n}$ . Choose secret  $P$  and independent  $D, \tilde{D}$ :

$$\text{public key: } y = \text{RREF}(G_0 PD), \quad \text{tag: } \tau = \text{RREF}(T_0 P \tilde{D}).$$

Link signatures by comparing their canonical tags:

$$\text{Link}(\sigma, \sigma') = 1 \iff \text{Can}(\tau) = \text{Can}(\tau').$$

| Assumption         | Intended meaning                                                        |
|--------------------|-------------------------------------------------------------------------|
| Non-splitting      | A key cannot support incompatible canonical tags.                       |
| 2-pseudorandomness | A shared representative is hidden in the public key/tag pair.           |
| Frame one-wayness  | An honest user's canonical tag cannot be forged into a fresh signature. |

1. Key and tag can use different secrets. They share  $\text{Rep}(g) = \text{Rep}(\tilde{g})$ , left compatibility preserves this under masking.
2. Effectiveness connects the new search problem to GAIP.
3. LRS security requires separate assumptions.

# Advertisement? in my slides??



Informal doctoral network for young researchers: online talks and events on codes and cryptography.



Call for interest.

[forms.gle/7Vz9gYn1S5bmqkZNA](https://forms.gle/7Vz9gYn1S5bmqkZNA)

Leonardo Errati  
[leonardo.errati@polito.it](mailto:leonardo.errati@polito.it)

Laura Mattiuz  
[l.mattiuz@pm.univpm.it](mailto:l.mattiuz@pm.univpm.it)

- Beullens, W., S. Katsumata, and F. Pintore (2020). “Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, pp. 464–492.
- Biasse, J.-F., G. Micheli, E. Persichetti, and P. Santini (2020). “LESS is More: Code-Based Signatures Without Syndromes”. In: *AFRICACRYPT 2020*, pp. 45–65.
- Budroni, A., J.-J. Chi-Domínguez, G. D’Alconzo, A. J. Di Scala, and M. Kulkarni (2024). “Don’t use it twice! Solving relaxed linear equivalence problems”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, pp. 35–65.
- Budroni, A., J.-J. Chi-Domínguez, and E. Franch (2025). “Don’t Use It Twice: Reloaded! On the Lattice Isomorphism Group Action”. In: *IACR Communications in Cryptology 2.2*, pp. 1–15.
- Budroni, A., G. D’Alconzo, L. Errati, and E. Persichetti (2026). *On Canonical Liftings, Group Actions, and Linkable Ring Signatures*. Cryptology ePrint Archive, Paper 2026/1348.

- Bultel, X. and C. Olivier-Anclin (2024). “On the Anonymity of Linkable Ring Signatures”. In: *CANS 2024*. <https://eprint.iacr.org/2025/131>.
- Duong, D. H., X. T. Khuc, Y. Qiao, W. Susilo, and C. Zhang (2026). “(Partially) Blind Signatures from Cryptographic Group Actions”. In: *Applied Cryptography and Network Security – ACNS 2026*. Vol. 16571. Lecture Notes in Computer Science. Springer, pp. 158–186.
- Iribarnegaray, I. D., V. Gregor, and F. L. Leal (2026). *A note on the Unsuitability of LIGA for Linkable Ring Signatures: The perils of non-commutativity*. Cryptology ePrint Archive, Paper 2026/671.
- koe, K. M. Alonso, and S. Noether (2020). *Zero to Monero*. getmonero.org library. Second edition. <https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf>.
- Liu, J. K., V. K. Wei, and D. S. Wong (2004). “Linkable spontaneous anonymous group signature for ad hoc groups”. In: *Australasian conference on information security and privacy*. Springer, pp. 325–335.

- Maheswaran, J., D. Jackowitz, D. I. Wolinsky, L. Wang, and B. Ford (2014). *Crypto-Book: Bootstrapping Privacy Preserving Online Identities from Social Networks*. arXiv:1406.4053.
- Moody, D., R. Perlner, A. Regenscheid, A. Robinson, and D. Cooper (Oct. 2026). *Transition to Post-Quantum Cryptography Standards*. NIST Internal Report 8547. National Institute of Standards and Technology.
- NIST (2026a). *Additional Digital Signature Schemes*. NIST CSRC. Consulted 25 October 2026. <https://csrc.nist.gov/projects/pqc-dig-sig>.

- NIST (2022). *PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates*. NIST News. First-round candidates: NIST IR 8240 (2019); second-round announcement (30 January 2019); third-round announcement (22 July 2020); selection (5 July 2022).  
<https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf>;  
<https://csrc.nist.gov/news/2019/pqc-standardization-process-2nd-round-candidates>; <https://csrc.nist.gov/news/2020/pqc-third-round-candidate-announcement>;  
<https://www.nist.gov/news-events/news/2022/07/pqc-standardization-process-announcing-four-candidates-be-standardized-plus>.
- NIST (2026b). *Nine Candidates Advance to the Third Round of the Additional Digital Signatures*. NIST News. 14 May 2026, updated 29 July 2026; consulted 26 October 2026.

Rivest, R. L., A. Shamir, and Y. Tauman (2001). "How to leak a secret". In: *International conference on the theory and application of cryptology and information security*. Springer, pp. 552–565.