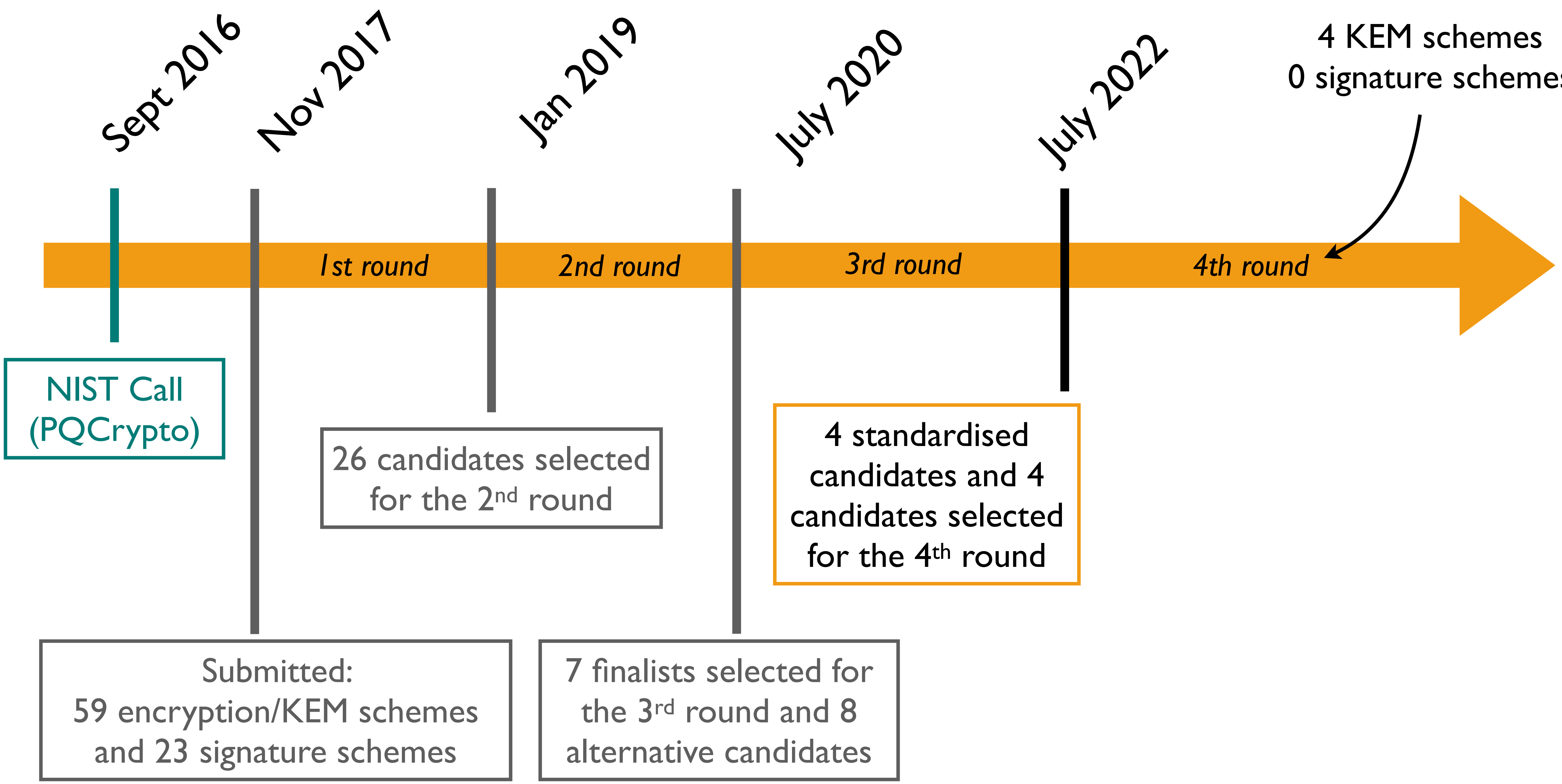
# Post-quantum Signatures from Secure Multiparty Computation

Thibauld Feneuil

June 14th, 2023 — WRACH'23

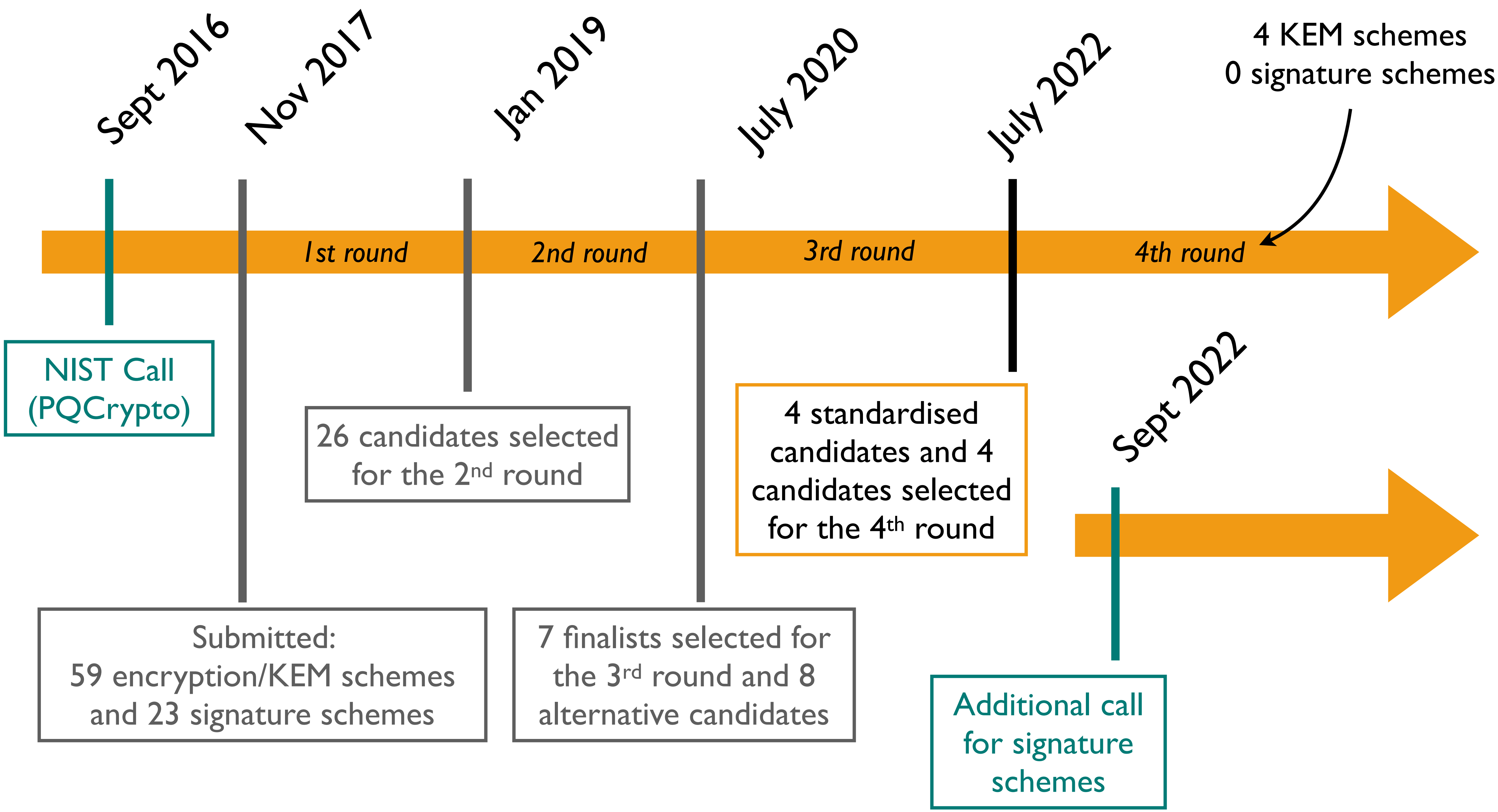
# Context

- Additional NIST call for quantum-resilient signature schemes



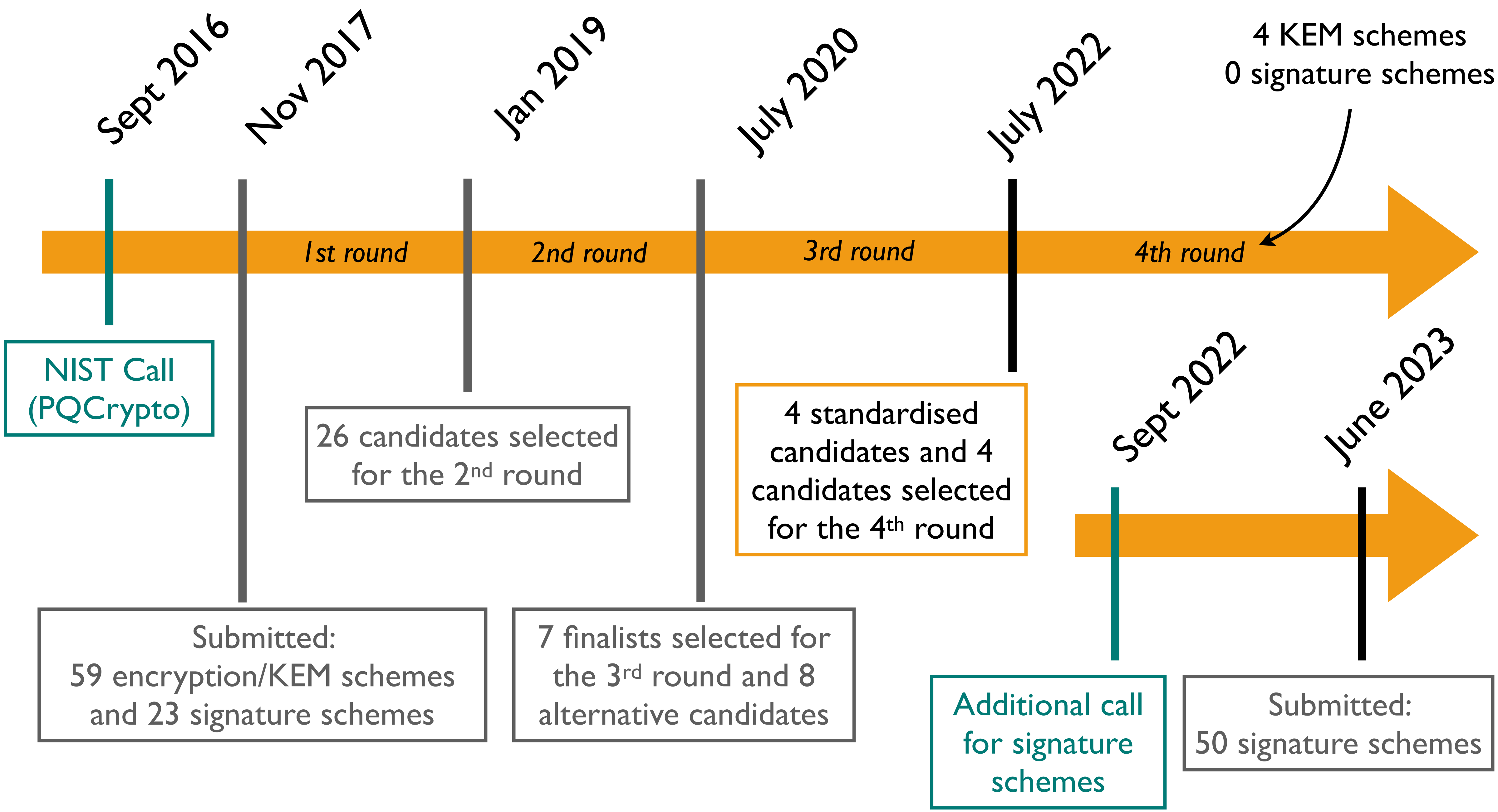
# Context

- Additional NIST call for quantum-resilient signature schemes



# Context

- Additional NIST call for quantum-resilient signature schemes



# Context

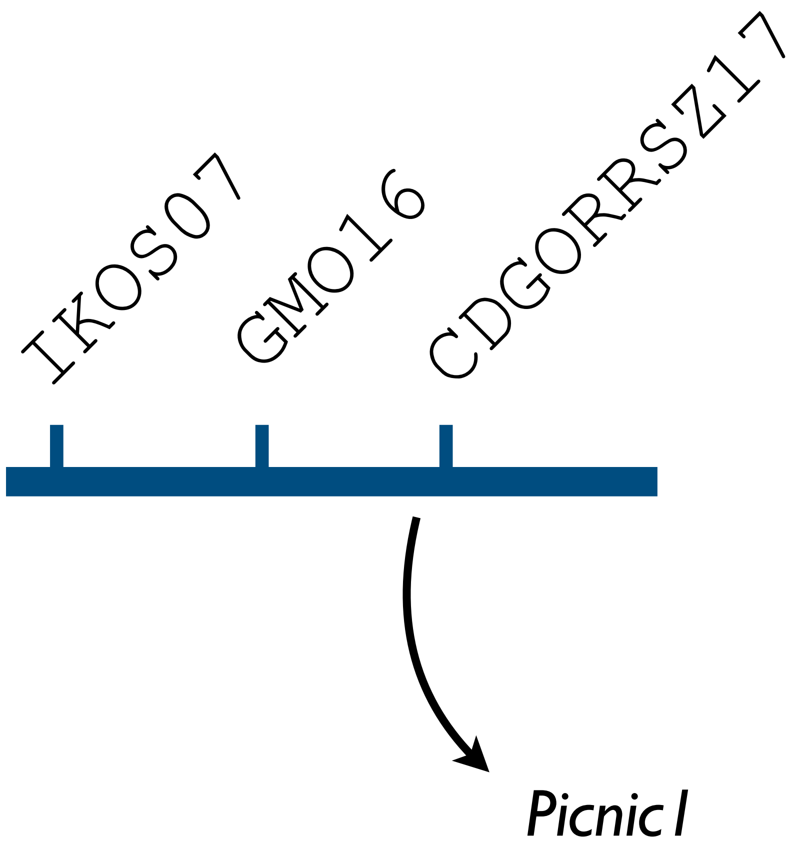
- Additional NIST call for quantum-resilient signature schemes

## CATEGORIES

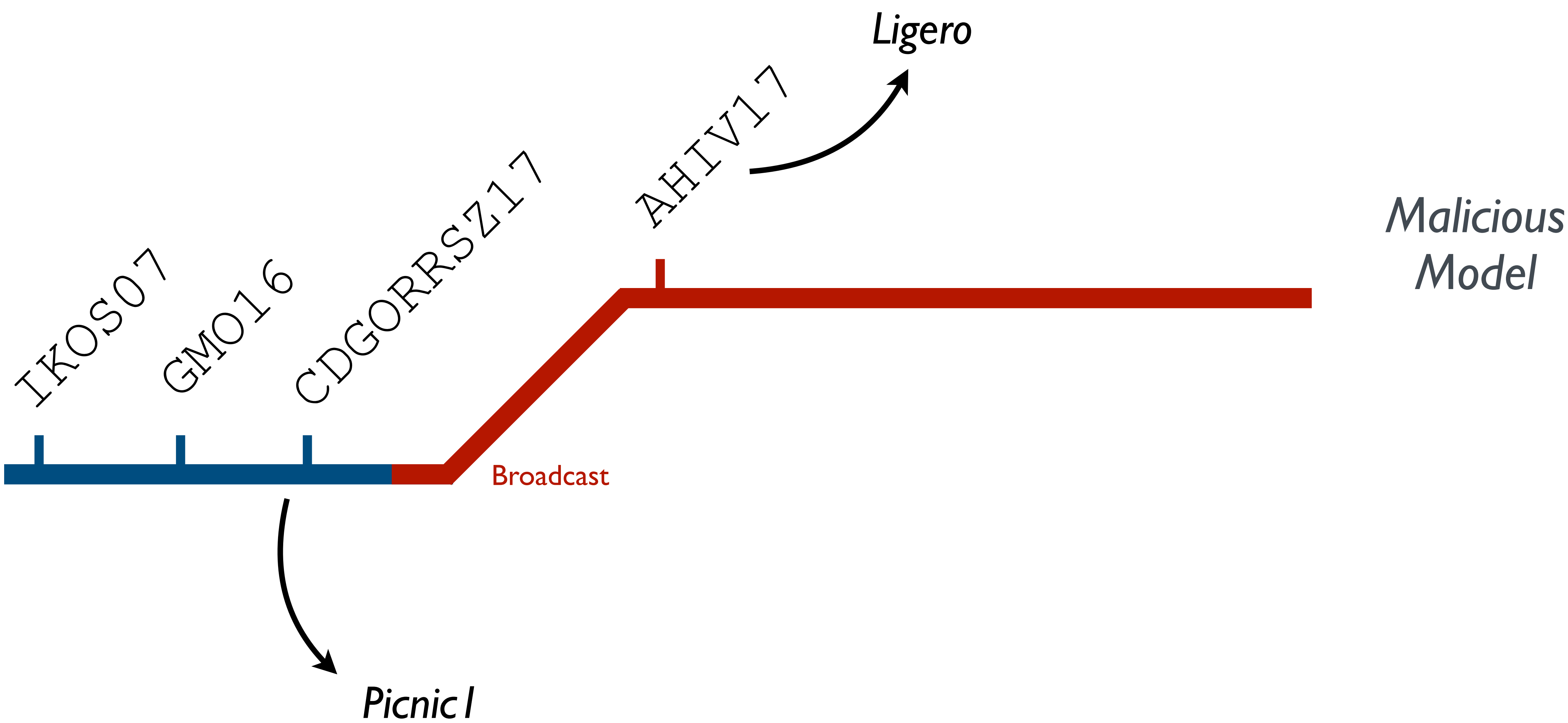
| Type | Number |
|---|---|
| Lattice | 8 |
| Code-based | 5 |
| Multivariate | 11 |
| MPC in the head | 7 |
| Symmetric | 6 |
| Isogeny | 1 |
| Other | 12 |
| Total | 50 |

*Source: NIST, 9th June 2023*

# MPC-in-the-Head Paradigm



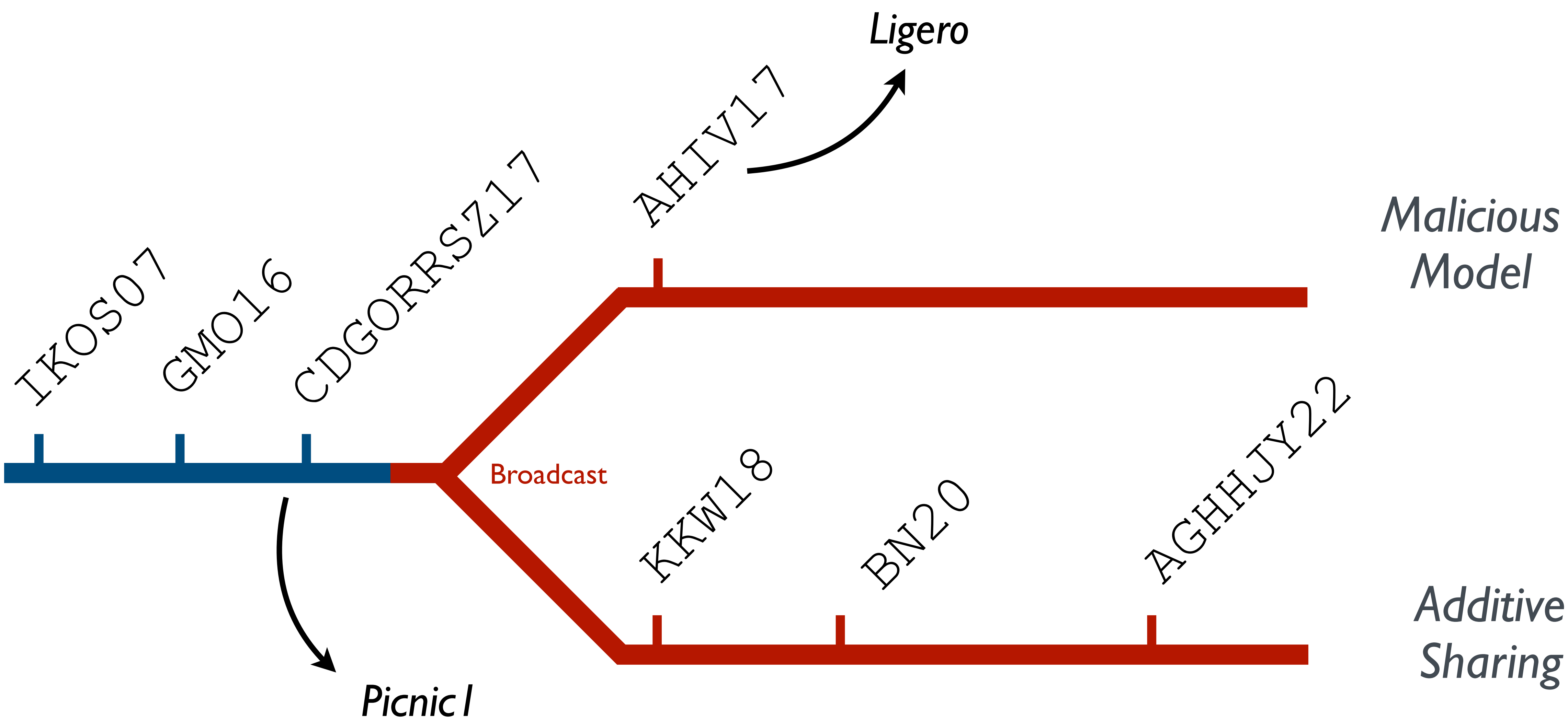
# MPC-in-the-Head Paradigm



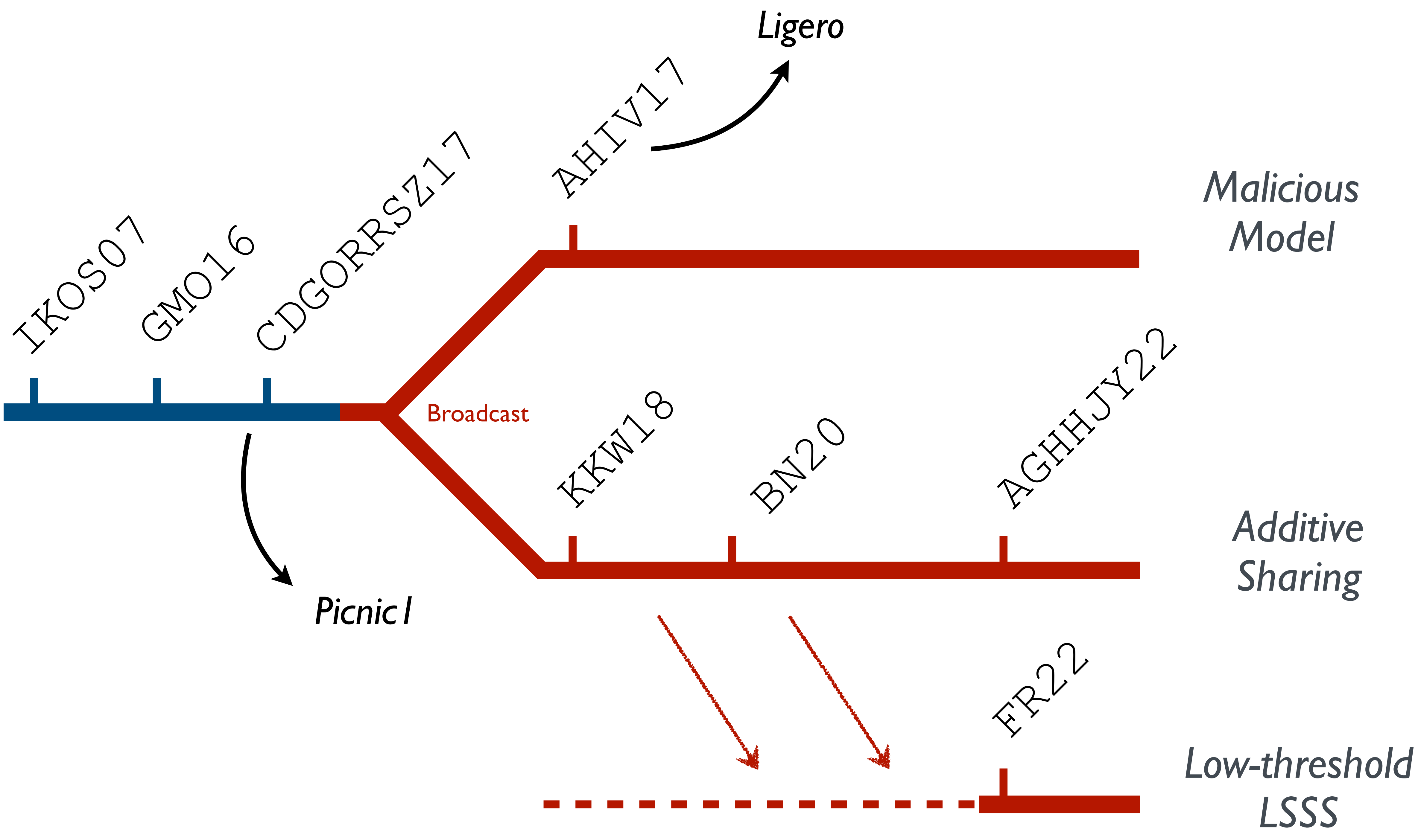



CryptoExperts

# MPC-in-the-Head Paradigm



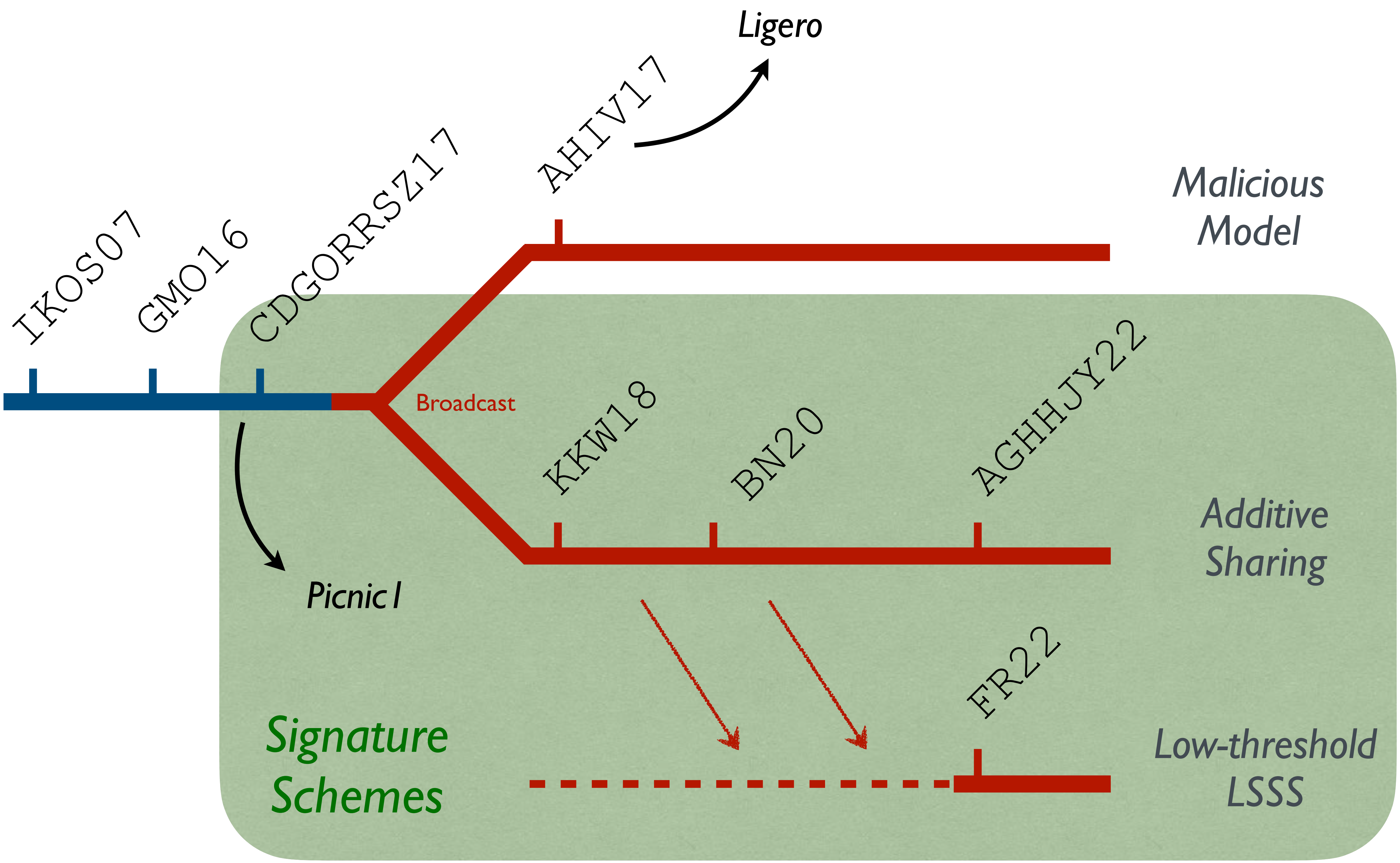
# MPC-in-the-Head Paradigm






CryptoExperts

# MPC-in-the-Head Paradigm



# Build a signature scheme from MPC

- Choose an one-way function $F$.

Methodology

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \le w$.

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.

# Build a signature scheme from MPC

[FJR22]

Methodology

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.

Example

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying

$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$

with $\deg Q = w$.

# Rephrase the pre-image verification

Let us assume that we have

$$Q(X) \cdot \underbrace{\left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right)}_{\substack{\text{is equal to } x_k \\ \text{when evaluating in } \gamma_k}} = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$

with $\deg Q = w$.

# Rephrase the pre-image verification

Let us assume that we have

$$Q(X)\cdot\underbrace{\left(\sum_{i=1}^{m} x_i \prod_{j=1,j\neq i}^{m}\frac{X-\gamma_j}{\gamma_i-\gamma_j}\right)}_{\substack{\text{is equal to } x_k\\ \text{when evaluating in } \gamma_k}} = P(X)\cdot\left(\prod_{i=1}^{m}(X-\gamma_i)\right)$$

with $\deg Q = w$.

Let us take $\gamma_k \in \{\gamma_1,\ldots,\gamma_m\}$,

$$Q(\gamma_k)\cdot x_k = 0$$



CryptoExperts

# Rephrase the pre-image verification

Let us assume that we have

$$Q(X) \cdot \underbrace{\left( \sum_{i=1}^{m} x_i \prod_{j=1, j\neq i}^{m} \frac{X-\gamma_j}{\gamma_i - \gamma_j} \right)}_{\text{is equal to } x_k \text{ when evaluating in } \gamma_k} = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$

with $\deg Q = w$.

Let us take $\gamma_k \in \{\gamma_1, \ldots, \gamma_m\}$,

$$Q(\gamma_k) \cdot x_k = 0$$

Can be zero for at most $w$ values

# Rephrase the pre-image verification

Let us assume that we have

$$Q(X) \cdot \underbrace{\left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right)}_{\substack{\text{is equal to } x_k \\ \text{when evaluating in } \gamma_k}} = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$

with $\deg Q = w$.

Let us take $\gamma_k \in \{\gamma_1, \ldots, \gamma_m\}$,

$$Q(\gamma_k) \cdot x_k = 0$$

Can be zero for <u>at most</u> $w$ values

Must be zero for at least $w$ coordinates ($w_H(x) \leq x$)

# Rephrase the pre-image verification

To get a valid polynomial $Q$, we can take

$$Q(X) := Q'(X) \cdot \prod_{i=1,\ x_i \neq 0}^{m} (X - \gamma_i)$$

$$Q(\gamma_k) \cdot x_k = 0$$

Can be zero for <u>at most</u> $w$ values

Must be zero for at least $w$ coordinates $(\mathrm{w}_H(x) \leq x)$

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying
$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$
with $\deg Q = w$.

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.
- Design a dedicated MPC protocol for the pre-image verification.

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying
$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$
with $\deg Q = w$.

# Design a MPC protocol for SD

We need to check that the secret $x$ satisfies

$$y = Hx \quad \text{and} \quad \mathrm{w}_H(x) \leq w\,.$$

# Design a MPC protocol for SD

We need to check that the secret $x$ satisfies

$$y = Hx \quad \text{and} \quad Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j \neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right).$$

Linear relation
⇒ Easy to compute in MPC

The MPC protocol will sample a random public point $r$ and evaluate the polynomial relation on this point.

*__Schwartz-Zippel Lemma__*:
If the polynomial relation is not satisfied, then the probability that it is true for a random point is *__small__*.

Finally, the MPC protocol just needs to check a quadratic term:
$Q(r) \cdot S(r) = P(r) \cdot F(r)$

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.
- Design a dedicated MPC protocol for the pre-image verification.

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying
$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j\neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$
with $\deg Q = w$.
- An MPC protocol which evaluates the above polynomial relation on a random point (Schwartz-Zippel).

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.
- Design a dedicated MPC protocol for the pre-image verification.
- Apply a MPC-in-the-Head transformation

*Example*

- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying
$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j\neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$
with $\deg Q = w$.
- An MPC protocol which evaluates the above polynomial relation on a random point (Schwartz-Zippel).
- *Result*: a zero-knowledge proof of knowledge for the syndrome decoding problem.

# Build a signature scheme from MPC

*Use only a broadcast channel*
*Perform only linear operations*

An MPC protocol
verifying an SD solution

*Additive sharing*
*[KKW18,AGHHJY22]*

Shorter signature size
Highly parallelizable
Slower signing time
Signing time ≈ verification time
Computational cost is mainly
due to symmetric primitive

*Low-threshold*
*Shamir's secret sharing*
*[FR22]*

Faster signing time
Highly parallelizable
Very fast verification
A bit simpler implementation
Larger signature size
Restriction # of parties
Computational cost is mainly
due to arithmetics

An zero-knowledge proof
of knowledge for the SD problem

# Build a signature scheme from MPC

[FJR22]

*Methodology*

- Choose an one-way function $F$.
- Rephrase the pre-image verification, *i.e.* the arithmetic circuit verifying that we have $y = F(x)$, to have a more MPC-friendly circuit.
- Design a dedicated MPC protocol for the pre-image verification.
- Apply a MPC-in-the-Head transformation
- Make the scheme non-interactive (Fiat-Shamir transformation)

*Example*

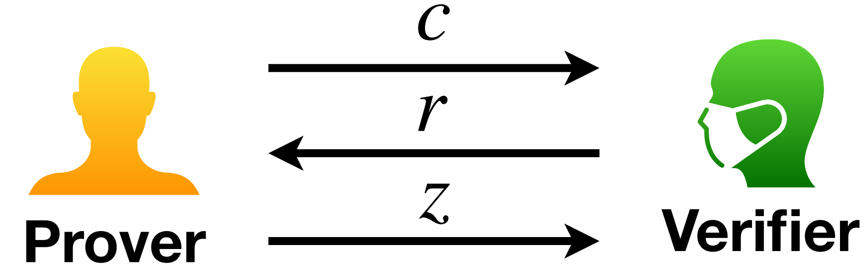
- *Syndrome decoding problem*: given $(H, y)$, find a vector $x$ such that $y = Hx$ and $\mathrm{w}_H(x) \leq w$.
- Find a vector $x$ such that $y = Hx$ and there exists two polynomials $Q$ and $P$ satisfying

$$Q(X) \cdot \left( \sum_{i=1}^{m} x_i \prod_{j=1, j\neq i}^{m} \frac{X - \gamma_j}{\gamma_i - \gamma_j} \right) = P(X) \cdot \left( \prod_{i=1}^{m} (X - \gamma_i) \right)$$

  with $\deg Q = w$.
- An MPC protocol which evaluates the above polynomial relation on a random point (Schwartz-Zippel).
- *Result*: a zero-knowledge proof of knowledge for the syndrome decoding problem.
- Result: a signature scheme relying on the syndrome decoding problem.

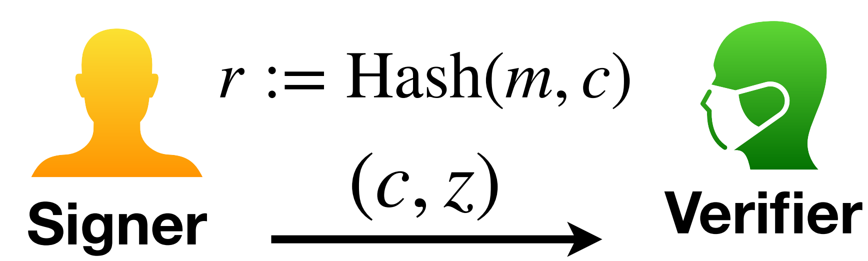
# Build a signature scheme from MPC

An zero-knowledge proof
of knowledge for the SD problem




*Fiat-Shamir Transformation*

A signature scheme relying on
the hardness of the SD problem



# Build a signature scheme from MPC

Choice of an
one-way function

*Arithmetization*

Circuit for a
pre-image verification

An MPC protocol
verifying an SD solution

*Additive sharing*

*Low-threshold*
*Shamir's secret sharing*

An zero-knowledge proof
of knowledge for the SD problem

*Fiat-Shamir Transformation*

A signature scheme relying on
the hardness of the SD problem

# An MPC-friendly statement

- *Size of the solution ring*: $$\text{size}_{bits} \geq \lambda^2 + \frac{\log_2 |\text{ring}|}{\log_2 N} \cdot \lambda$$
  - Lattice (SIS): ring of $2^{65\,536}$ solution candidates $\Rightarrow$ $\text{size}_{bits} \geq 133\ KB$
  - Code (SD): ring of $2^{1280}$ solution candidates $\Rightarrow$ $\text{size}_{bits} \geq 4.6\ KB$

# An MPC-friendly statement

- *Size of the solution ring*: $$\text{size}_{bits} \geq \lambda^2 + \frac{\log_2 |\text{ring}|}{\log_2 N} \cdot \lambda$$
  - Lattice (SIS): ring of $2^{65\,536}$ solution candidates $\Rightarrow$ size$_{bits} \geq 133\ KB$
  - Code (SD): ring of $2^{1280}$ solution candidates $\Rightarrow$ size$_{bits} \geq 4.6\ KB$

- *Size of the base field:* the current MPC techniques for MPCitH are more efficient with large fields (for example, the Schwartz-Zippel Lemma).
  - SD over $GF(2)$: around 11-13 KB
  - SD over $GF(256)$: around 8-9 KB

# An MPC-friendly statement

- *Size of the solution ring*: $$\text{size}_{bits} \geq \lambda^2 + \frac{\log_2 |\text{ring}|}{\log_2 N} \cdot \lambda$$
  - Lattice (SIS): ring of $2^{65\,536}$ solution candidates $\Rightarrow$ size$_{bits} \geq 133\ KB$
  - Code (SD): ring of $2^{1280}$ solution candidates $\Rightarrow$ size$_{bits} \geq 4.6\ KB$
- *Size of the base field:* the current MPC techniques for MPCitH are more efficient with large fields (for example, the Schwartz-Zippel Lemma).
  - SD over $GF(2)$: around 11-13 KB
  - SD over $GF(256)$: around 8-9 KB
- *Multiplicative depth of the verification circuits*
  - Having a depth of $1$ is the optimal.
  - SD over GF(256): depth of $1$, around 8-9 KB
  - PKP: depth of $\log_2 n$, around 12-13 KB

# An MPC-friendly statement

- *Size of the solution ring*: $\text{size}_{bits} \geq \lambda^2 + \dfrac{\log_2 |\text{ring}|}{\log_2 N} \cdot \lambda$
  - Lattice (SIS): ring of $2^{65\,536}$ solution candidates $\Rightarrow$ $\text{size}_{bits} \geq 133\ KB$
  - Code (SD): ring of $2^{1280}$ solution candidates $\Rightarrow$ $\text{size}_{bits} \geq 4.6\ KB$
- *Size of the base field:* the current MPC techniques for MPCitH are more efficient with large fields (for example, the Schwartz-Zippel Lemma).
  - SD over $GF(2)$: around 11-13 KB
  - SD over $GF(256)$: around 8-9 KB
- *Multiplicative depth of the verification circuits*
  - Having a depth of $1$ is the optimal.
  - SD over GF(256): depth of $1$, around 8-9 KB
  - PKP: depth of $\log_2 n$, around 12-13 KB
- Number of multiplications in the verification circuit

# Signature scheme: SD-in-the-Head

- Many (standard) MPCitH optimisations to reduce the signature size
- Obtained signature sizes:

| Field | PK size | Signature Size | |
|---|---|---|---|
| | | Additive | LSSSitH |
| GF(2) | 90-100 B | 11-13 KB | - |
| GF(251) | 140-150 B | 8-9 KB | 9.5-10.5 KB |
| GF(256) | 140-150 B | 8-9 KB | 9.5-10.5 KB |

# Signature scheme: SD-in-the-Head

- Many (standard) MPCitH optimisations to reduce the signature size
- Obtained signature sizes:

| Field | PK size | Signature Size | |
|---|---|---|---|
| | | Additive | LSSSitH |
| GF(2) | 90-100 B | 11-13 KB | - |
| GF(251) | 140-150 B | 8-9 KB | 9.5-10.5 KB |
| GF(256) | 140-150 B | 8-9 KB | 9.5-10.5 KB |

*Why GF(251) or GF(256)?*

# Signature scheme: SD-in-the-Head

- Many (standard) MPCitH optimisations to reduce the signature size
- Obtained signature sizes:

| Field | PK size | Signature Size | |
|---|---|---|---|
| | | Additive | LSSSitH |
| GF(2) | 90-100 B | 11-13 KB | - |
| GF(251) | 140-150 B | 8-9 KB | 9.5-10.5 KB |
| GF(256) | 140-150 B | 8-9 KB | 9.5-10.5 KB |

## *Why GF(251) or GF(256)?*

- Additive: the computational bottleneck is the pseudo-random generation (and the commitments). GF(256) will be more efficient than GF(251)
- LSSSitH: the computational bottleneck is the arithmetics. GF(251) will be more efficient than GF(256), especially on platforms without GFNI.

# Performances

*NIST Candidate SD-in-the-Head: Benchmark on a 2.60GHz recent platform*

| | Additive Sharing | | | LSSSitH | | |
|---|---|---|---|---|---|---|
| | Size | Sign | Verify | Size | Sign | Verify |
| SDitH (256) | **8 241** | **5.18** | **4.81** | 10 117 | 1.97 | 0.62 |
| SDitH (251) | 8 241 | 8.51 | 8.16 | **10 117** | **1.71** | **0.23** |

*Size in bytes, timing in milliseconds*

# Submitted candidates at NIST call

- *Syndrome Decoding Problem:*

**SD-in-the-Head**

C. Aguilar Melchor, T. Feneuil, N. Gama, S. Gueron, J. Howe, D. Joseph,
A. Joux, E. Persichetti, T. Randrianarisoa, M. Rivain, D. Yue.

- *Rank Syndrome Decoding Problem:*

**RYDE**

N. Aragon, M. Bardet, L. Bidoux, J.-J. Chi-Domínguez, V. Dyseryn,
T. Feneuil, P. Gaborit, A. Joux, M. Rivain, J.-P. Tillich, A. Vinçotte.

- *Min Rank Problem:*

**MIRA**

N. Aragon, M. Bardet, L. Bidoux, J.-J. Chi-Domínguez, V. Dyseryn,
T. Feneuil, P. Gaborit, R. Neveu, M. Rivain, J.-P. Tillich.

- *Multivariate Quadratic Problem:*

**MQOM: MQ on my Mind**

T. Feneuil, M. Rivain

# Submitted candidates at NIST call

- *How to deal with the rank metric [Fen22]:*
  - *Technique 1: let us have a matrix* $X \in \mathbb{F}_q^{n\times m}$

$$\mathrm{rk}(X) \leq r \iff \exists T \in \mathbb{F}^{n\times r}, R \in \mathbb{F}^{r\times m} : X = T \cdot R$$

  - *Technique 2: let us have a vector* $x \in \mathbb{F}_{q^m}^n$

$$\mathrm{w}_R(x) \leq r \iff \exists P(X) := X^{q^r} + \sum_{j=0}^{r-1} \beta_j X^{q^j} : \forall i : P(x_j) = 0$$

# Submitted candidates at NIST call

- *How to deal with the rank metric [Fen22]:*
  - *Technique 1: let us have a matrix* $X \in \mathbb{F}_q^{n\times m}$

$$\mathrm{rk}(X) \leq r \iff \exists T \in \mathbb{F}^{n\times r}, R \in \mathbb{F}^{r\times m} : X = T \cdot R$$

*Lighter scheme*

  - *Technique 2: let us have a vector* $x \in \mathbb{F}_{q^m}^n$

$$\mathrm{w}_R(x) \leq r \iff \exists P(X) := X^{q^r} + \sum_{j=0}^{r-1} \beta_j X^{q^j} : \forall i : P(x_j) = 0$$

*Shorter size*

# Submitted candidates at NIST call

- *How to deal with the rank metric [Fen22]:*
  - *Technique 1: let us have a matrix $X \in \mathbb{F}_q^{n\times m}$*

$$\mathrm{rk}(X) \leq r \iff \exists T \in \mathbb{F}^{n\times r}, R \in \mathbb{F}^{r\times m} : X = T \cdot R$$

Lighter scheme

  - *Technique 2: let us have a vector $x \in \mathbb{F}_{q^m}^n$*

$$\mathrm{w}_R(x) \leq r \iff \exists P(X) := X^{q^r} + \sum_{j=0}^{r-1} \beta_j X^{q^j} : \forall i : P(x_j) = 0$$

Shorter size

- *Rank SD: Technique 2 is the best*
  - *From $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $y \in \mathbb{F}_q^{n-k}$, find a vector $x \in \mathbb{F}_{q^m}^n$ such that $y = Hx$ and $\mathrm{w}_R(x) \leq r$.*
- *Min Rank: not clear which technique is the best*
  - *From $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{m\times n}$,*

    *find a vector $x \in \mathbb{F}_q^k$ such that $\mathrm{rk}(M_0 + \sum_{i=1}^{k} x_i M_i) \leq r$.*

# Submitted candidates at NIST call

- *How to deal with the rank metric [Fen22]:*
  - *Technique 1: let us have a matrix* $X \in \mathbb{F}_q^{n\times m}$

$$\mathrm{rk}(X) \leq r \iff \exists T \in \mathbb{F}^{n\times r}, R \in \mathbb{F}^{r\times m} : X = T \cdot R$$

Lighter scheme

  - *Technique 2: let us have a vector* $x \in \mathbb{F}_{q^m}^n$

$$\mathrm{w}_R(x) \leq r \iff \exists P(X) := X^{q^r} + \sum_{j=0}^{r-1} \beta_j X^{q^j} : \forall i : P(x_j) = 0$$

Shorter size

- *Rank SD: Technique 2 is the best*
  - *From* $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ *and* $y \in \mathbb{F}_q^{n-k}$*, find a vector* $x \in \mathbb{F}_{q^m}^n$ *such that* $y = Hx$ *and* $\mathrm{w}_R(x) \leq r$.
- *Min Rank: not clear which technique is the best*
  - *From* $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{m\times n}$,

    *find a vector* $x \in \mathbb{F}_q^k$ *such that* $\mathrm{rk}(M_0 + \sum_{i=1}^{k} x_i M_i) \leq r$.

Technique 1: MiRith
Technique 2: MIRA

# Performances

*NIST Candidates: Benchmark on a 2.60GHz recent platform*

| | Additive Sharing | | | LSSSitH | | |
|---|---|---|---|---|---|---|
| | Size | Sign | Verify | Size | Sign | Verify |
| SDitH (256) | 8 241 | 5.18 | 4.81 | 10 117 | 1.97 | 0.62 |
| SDitH (251) | 8 241 | 8.51 | 8.16 | 10 117 | 1.71 | 0.23 |
| MQOM (31) | 6 348 | 17.06 | 16.05 | - | - | - |
| MQOM (251) | 6 575 | 10.97 | 10.50 | ~ 14 000 | - | - |
| RYDE | 5 956 | 8.58 | 7.31 | ~ 9 200 | - | - |
| MIRA (16) | 5 640 | 16.65 | 15.61 | - | - | - |

*Size in bytes, timing in milliseconds*
*Isochronous implementations*
*Single thread*

# Performances

*NIST Candidates: Benchmark on a 2.60GHz recent platform*

| | Additive Sharing | | | LSSSitH | | |
|---|---|---|---|---|---|---|
| | Size | Sign | Verify | Size | Sign | Verify |
| SDitH (256) | 8 241 | 5.18 | 4.81 | 10 117 | 1.97 | 0.62 |
| SDitH (251) | 8 241 | 8.51 | 8.16 | 10 117 | 1.71 | 0.23 |
| MQOM (31) | 6 348 | 17.06 | 16.05 | - | - | - |
| MQOM (251) | 6 575 | 10.97 | 10.50 | ~ 14 000 | - | - |
| RYDE | 5 956 | 8.58 | 7.31 | ~ 9 200 | - | - |
| MIRA (16) | 5 640 | 16.65 | 15.61 | - | - | - |

- Dilithium: \|sig\|=2420, \|pk\|=1312, $t_{sign}$=0.13, $t_{verify}$=0.05
- Falcon: \|sig\|=666, \|pk\|=897, $t_{sign}$=0.20, $t_{verify}$=0.06
- SPHINCS+: \|sig\|=7856, \|pk\|=32, $t_{sign}$=331, $t_{verify}$=2.3
  \|sig\|=17088, \|pk\|=32, $t_{sign}$=19, $t_{verify}$=0.9

# Conclusion

- MPC-in-the-Head
  - A practical tool to build *conservative* signature schemes
  - Very versatile and tunable
  - Can be applied on any one-way function

- Perspectives
  - *Additive-based MPCitH:* stable
  - *Low-threshold-based MPCitH*: new approach, could lead to follow-up works