Building MPCitH-based Signatures from MQ, MinRank and Rank SD

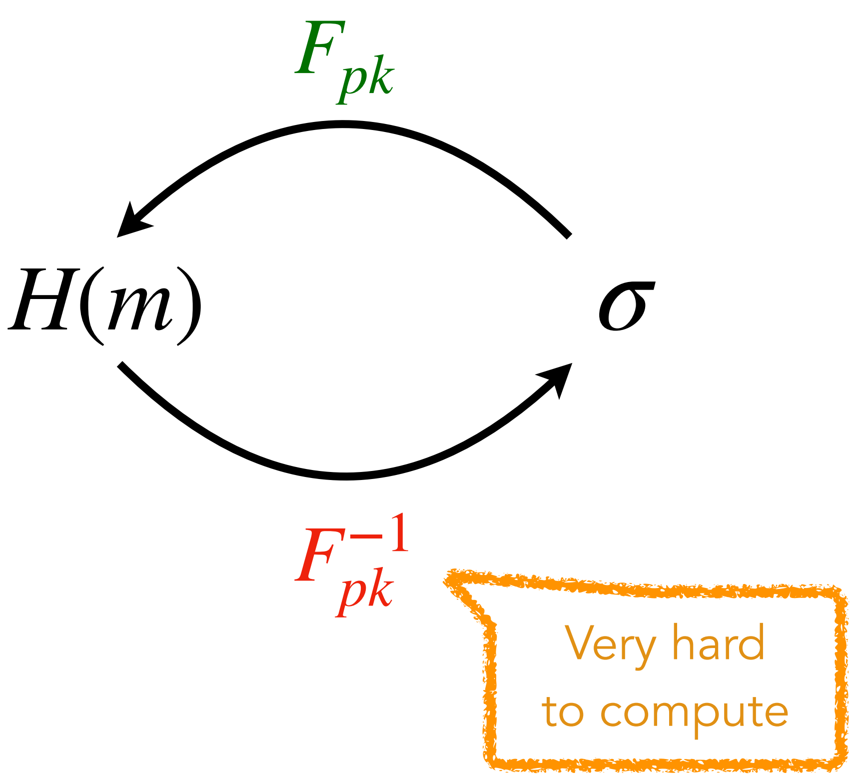
Thibauld Feneuil

ACNS'24

March 6, 2024 — Abu Dhabi (UAE)

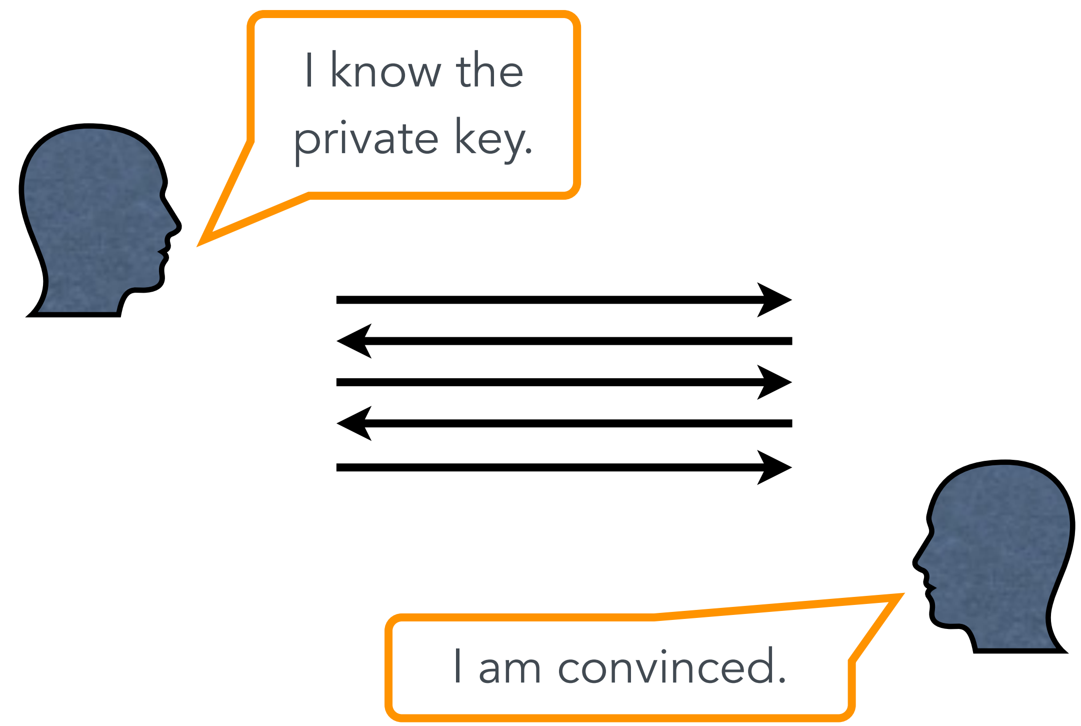
How to build signature schemes?

Hash & Sign



- Short signatures
- "Trapdoor" in the public key

From an identification scheme



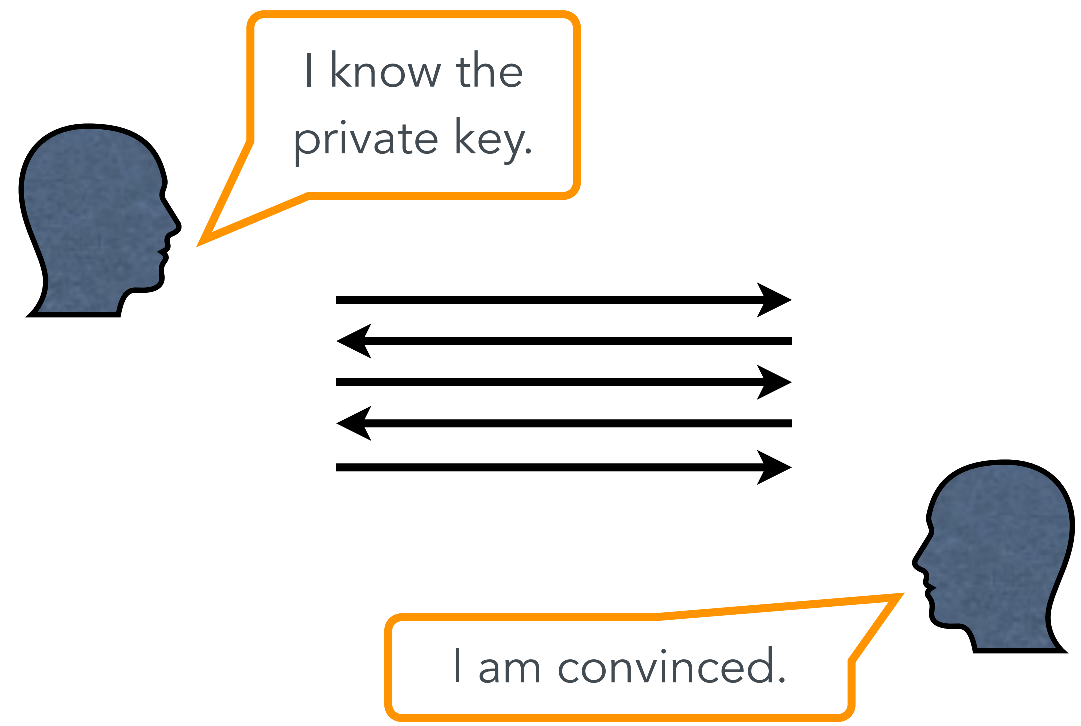
- Large(r) signatures
- Short public key

How to build signature schemes?

Hash & Sign

- Short signatures
- "Trapdoor" in the public key

From an identification scheme

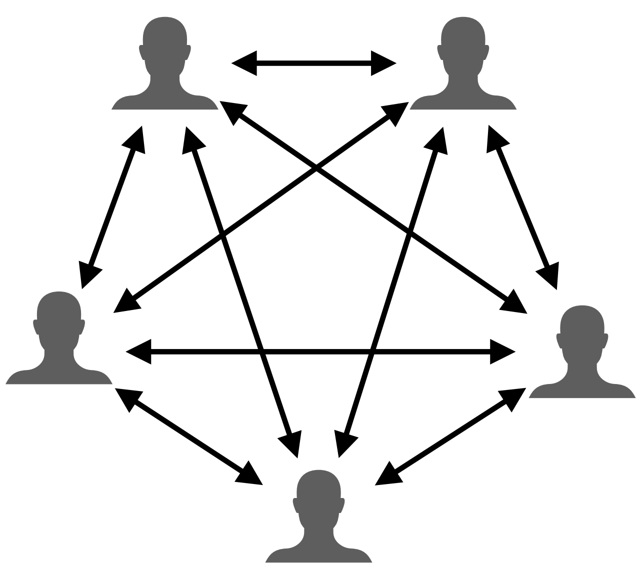


- Large(r) signatures
- Short public key

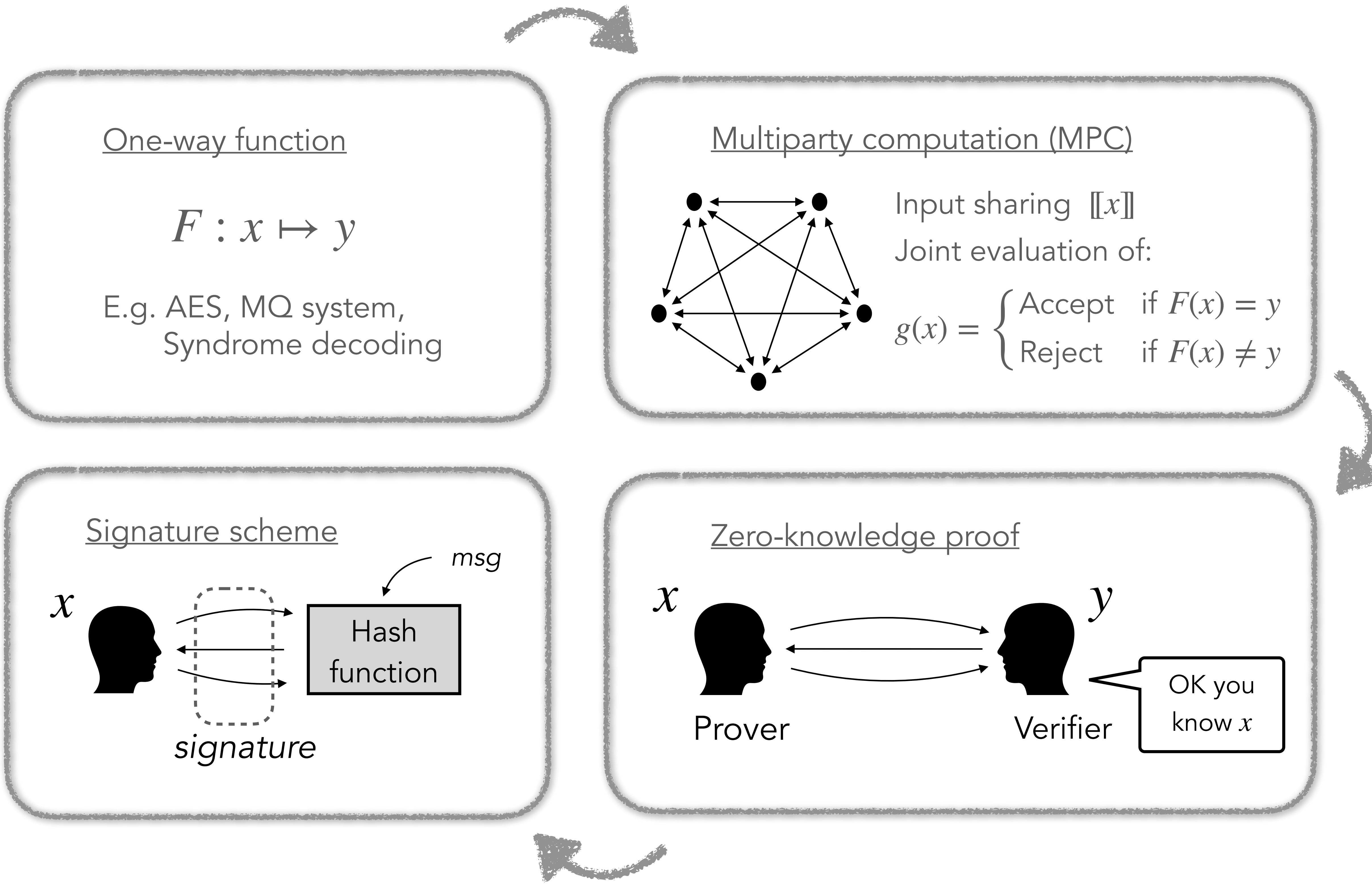
MPC in the Head

- **[IKOS07]** Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Amit Sahai: "Zero-knowledge from secure multiparty computation" (STOC 2007)

- Turn a *multiparty computation* (MPC) into an identification scheme



- **Generic**: can be apply to any cryptographic problem



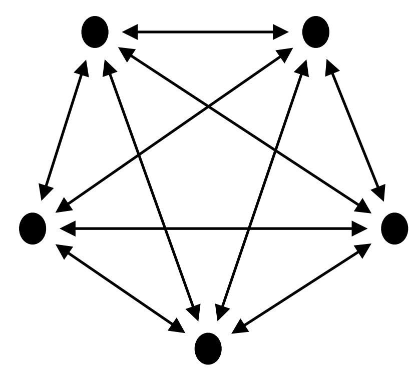
One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Zero-knowledge proof
x
y
Prover
Verifier
OK you know x
Signature scheme
x
msg
Hash function
signature

One-way function

$$F : x \mapsto y$$

E.g. AES, MQ system,
Syndrome decoding

Multiparty computation (MPC)

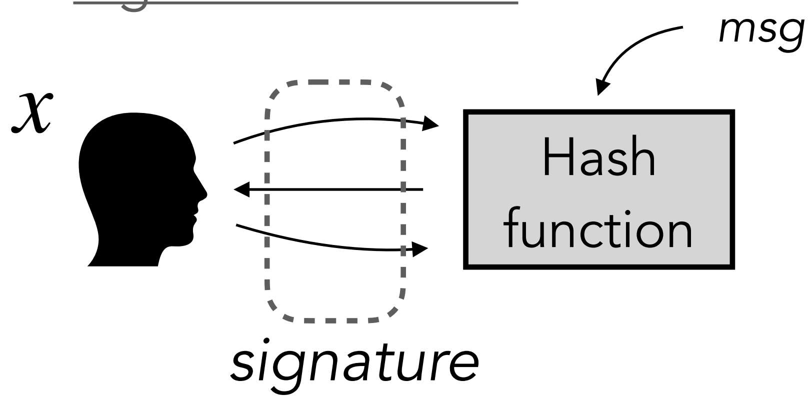


Input sharing $[\![x]\!]$

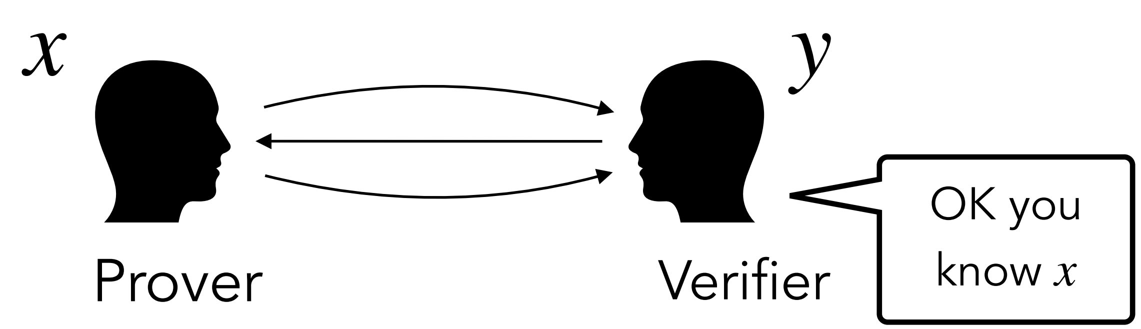
Joint evaluation of:

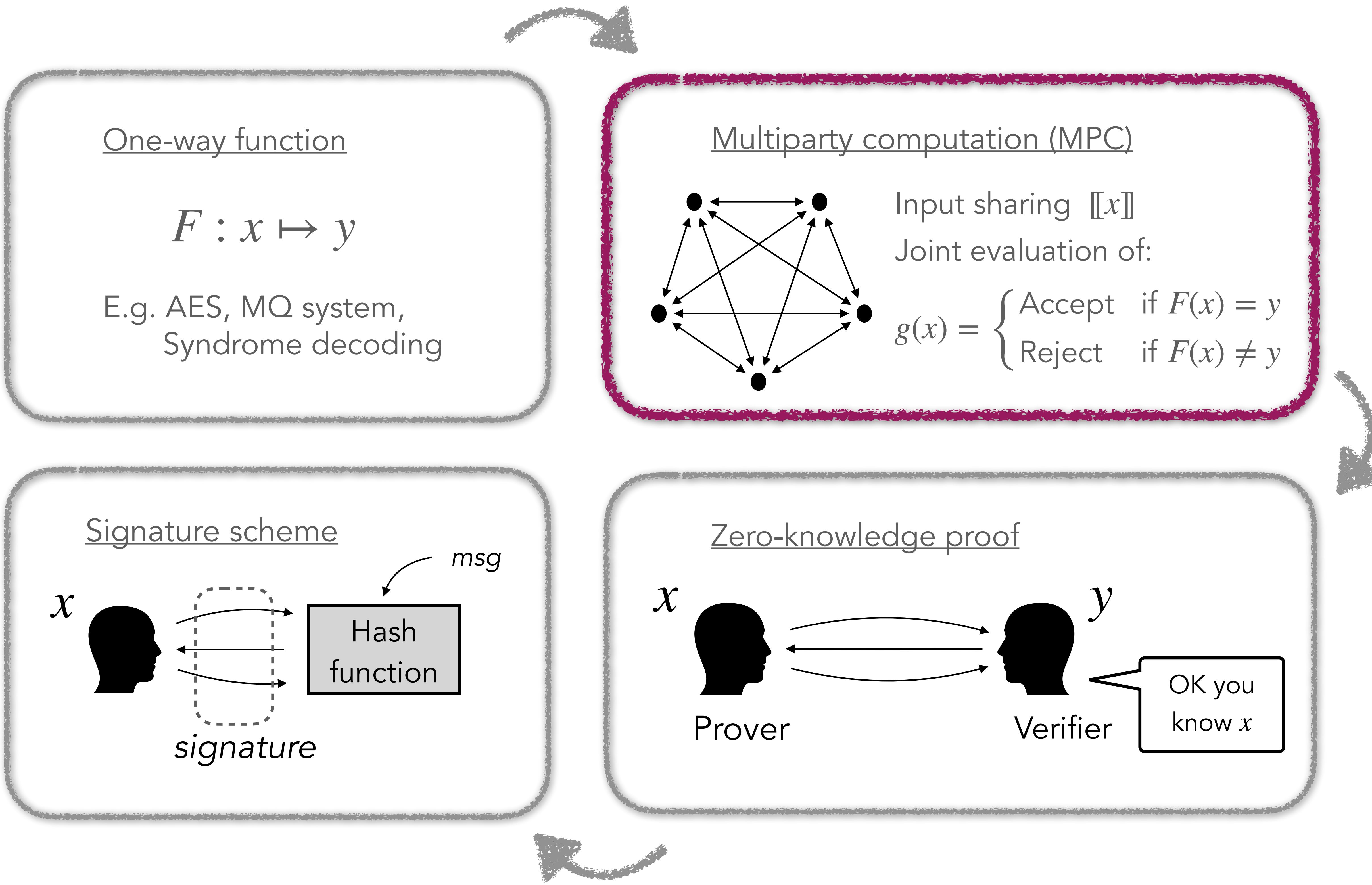
$$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$$

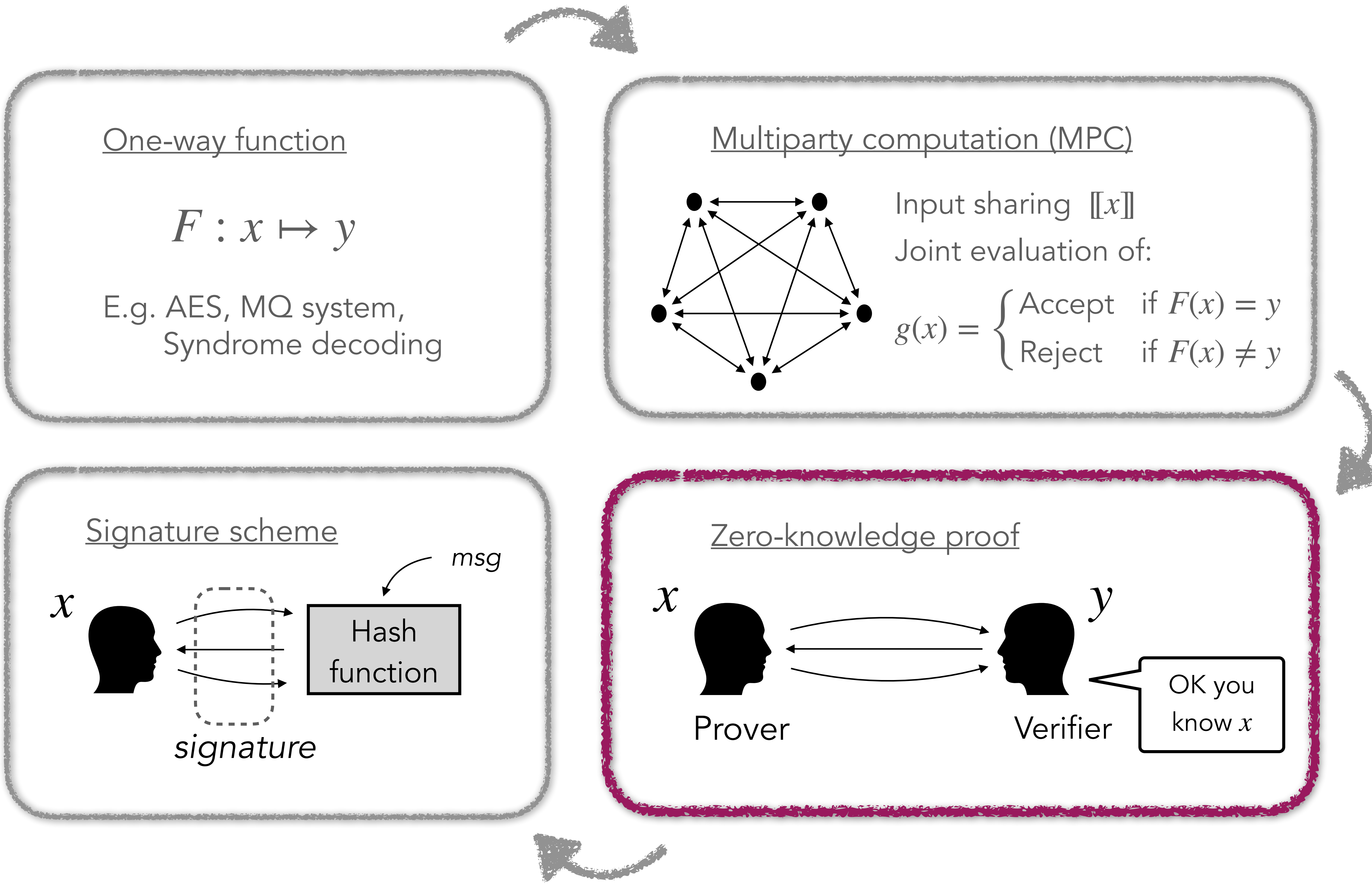
Signature scheme



Zero-knowledge proof




One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Signature scheme
x
msg
Hash function
signature
Zero-knowledge proof
x
y
Prover
Verifier
OK you know x

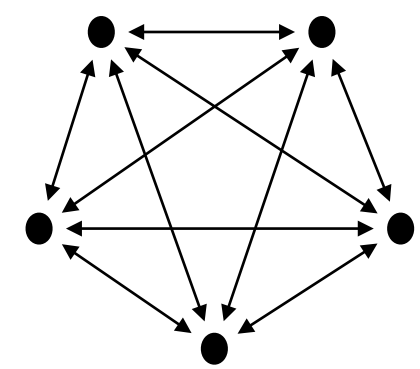

One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Signature scheme
x
msg
Hash function
signature
Zero-knowledge proof
x
y
Prover
Verifier
OK you know x

One-way function

$$F : x \mapsto y$$

E.g. AES, MQ system,
Syndrome decoding

Multiparty computation (MPC)

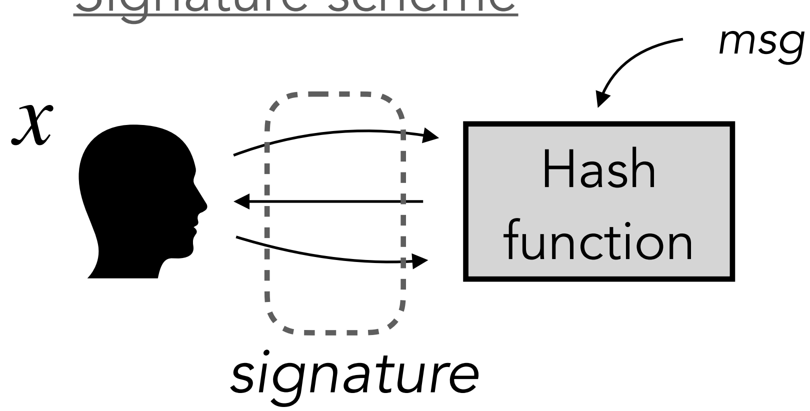


Input sharing $[\![x]\!]$

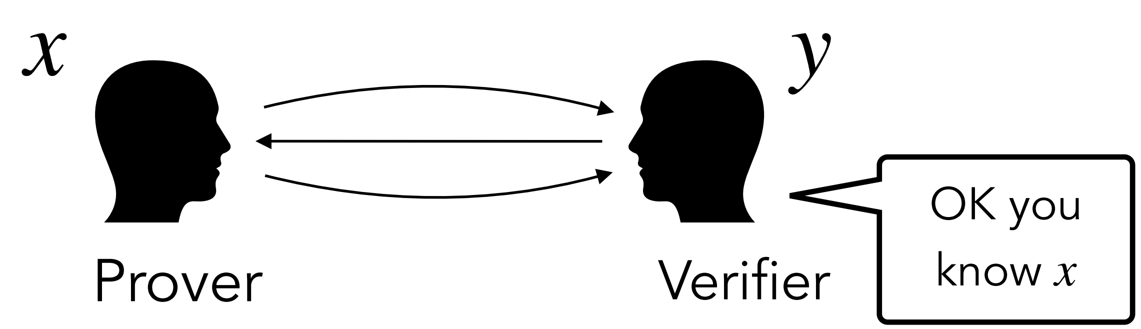
Joint evaluation of:

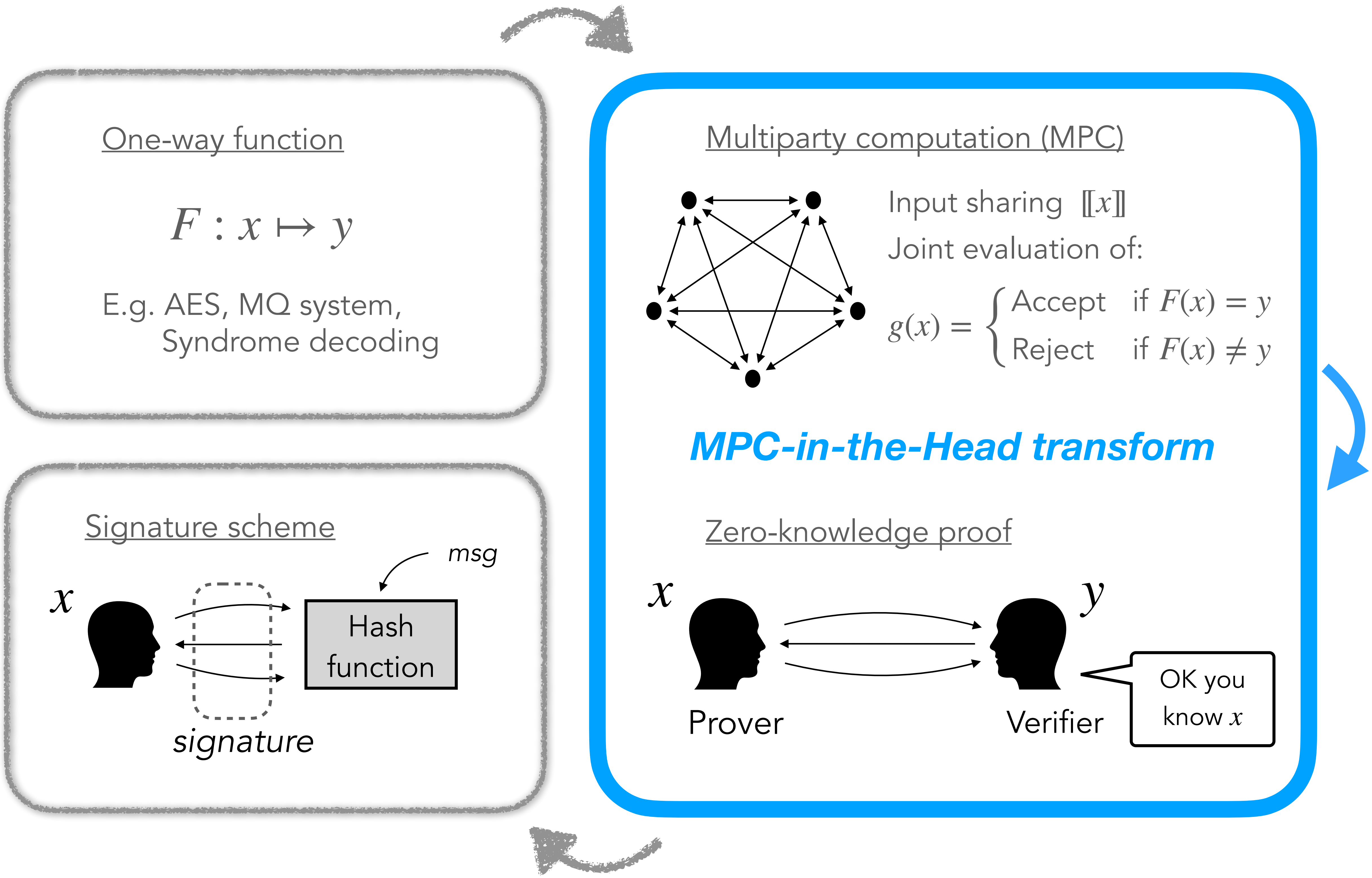
$$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$$

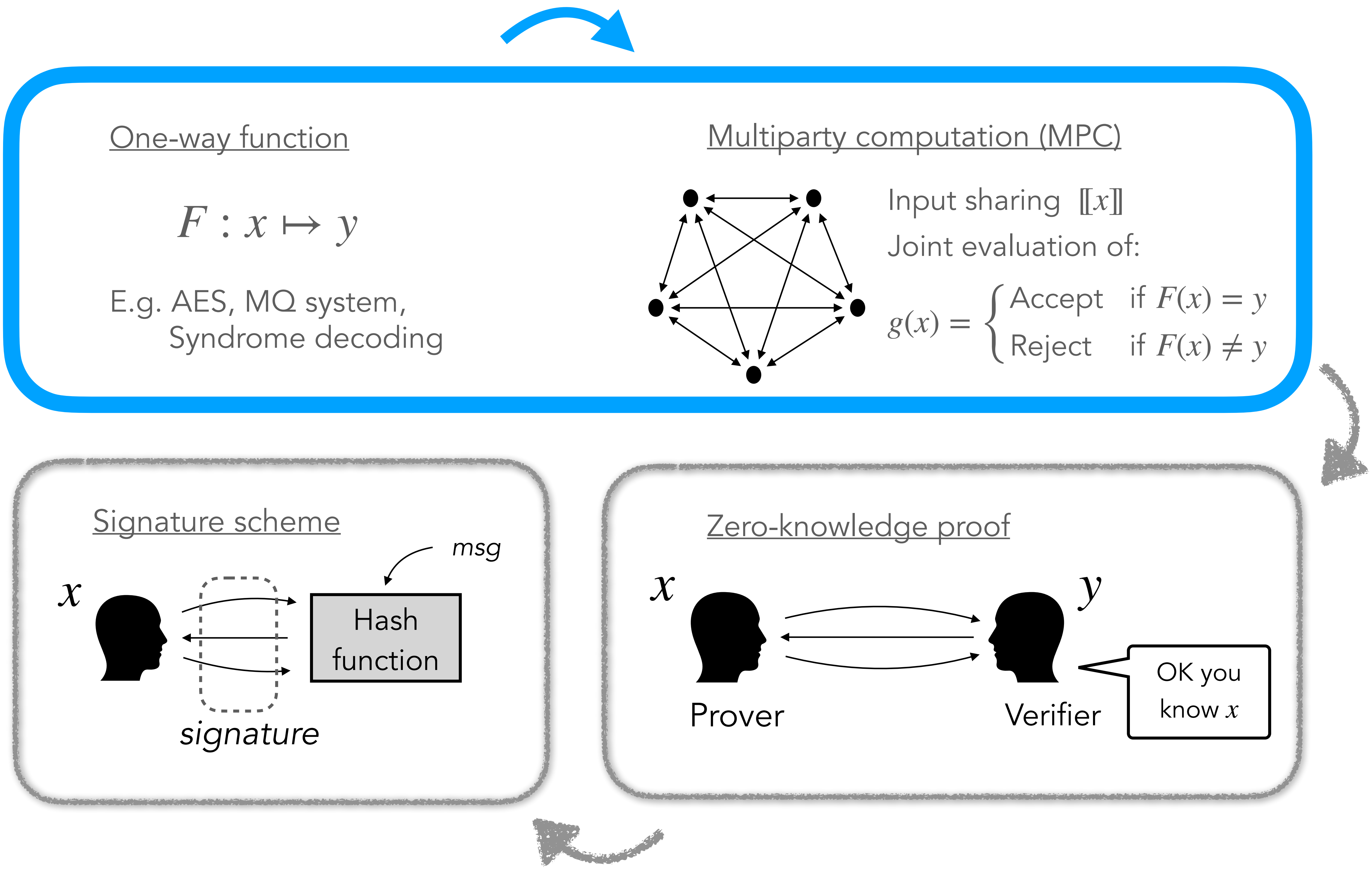
Signature scheme



Zero-knowledge proof




One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Signature scheme
x
msg
Hash function
signature
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
MPC-in-the-Head transform
Zero-knowledge proof
x
y
Prover
Verifier
OK you know x



One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Signature scheme
x
msg
Hash function
signature
Zero-knowledge proof
x
y
Prover
Verifier
OK you know x

Designing the MPC protocol

- We consider only broadcast communication [KKW18] and linear operations.

- To minimize the signature size, we need to
  - Minimize the size of the input of the MPC protocol,
  - Minimize the size of the broadcasted values.

- Relax the MPC functionality [BN20].
  - If $F(x) = y$, the MPC protocol should always output `Accept`.
  - If $F(x) \neq y$, the MPC protocol should output `Reject` with high probability.

[KKW18] Katz, Kolesnikov, Wang: "Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures" (CCS 2018)

[BN20] Baum, Nof: "Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography" (PKC 2020)

MPC protocol for MQ

Multivariate Quadratic Problem

From $(A_1, \ldots, A_m, b_1, \ldots, b_m, y_1, \ldots, y_m)$, find $x \in \mathbb{F}_q^n$ such that

$$\forall i \leq m, \quad y_i = x^T A_i x + b_i^T x.$$

The multi-party computation must check that the vector x satisfies

$$\begin{aligned} y_1 &= x^T A_1 x + b_1^T x \\ y_1 &= x^T A_1 x + b_1^T x \\ &\vdots \\ y_m &= x^T A_m x + b_m^T x \end{aligned}$$

MPC protocol for MQ

Multivariate Quadratic Problem

From $(A_1, \ldots, A_m, b_1, \ldots, b_m, y_1, \ldots, y_m)$, find $x \in \mathbb{F}_q^n$ such that

$$\forall i \leq m, \quad y_i = x^T A_i x + b_i^T x.$$

The multi-party computation must check that the vector x satisfies

$$\sum_{i=1}^{m} \gamma_i \cdot \left(y_i - x^T A_i x - b_i^T x\right) = 0$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

False positive rate:

$$\frac{1}{|\mathbb{F}_{ext}|}$$

MPC protocol for MQ

Multivariate Quadratic Problem

From $(A_1, \ldots, A_m, b_1, \ldots, b_m, y_1, \ldots, y_m)$, find $x \in \mathbb{F}_q^n$ such that

$$\forall i \leq m, \quad y_i = x^T A_i x + b_i^T x.$$

The multi-party computation must check that the vector x satisfies

$$\sum_{i=1}^{m} \gamma_i \cdot \left(y_i - b_i^T x\right) = \langle x, \left(\sum_{i=1}^{m} \gamma_i \cdot A_i\right) x \rangle$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

False positive rate:

$$\frac{1}{|\mathbb{F}_{ext}|}$$

MPC protocol for MQ

Multivariate Quadratic Problem

From $(A_1, \ldots, A_m, b_1, \ldots, b_m, y_1, \ldots, y_m)$, find $x \in \mathbb{F}_q^n$ such that

$$\forall i \leq m, \quad y_i = x^T A_i x + b_i^T x.$$

The multi-party computation must check that the vector x satisfies

$$\boxed{\sum_{i=1}^{m} \gamma_i \cdot \left(y_i - b_i^T x\right)} = \left\langle x, \boxed{\left(\sum_{i=1}^{m} \gamma_i \cdot A_i\right) x} \right\rangle$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

Linear into the secret values

False positive rate:

$$\frac{1}{|\mathbb{F}_{ext}|}$$

Signature Schemes from MQ

$q = 4$
$m = 88$
$n = 88$

	Variant	Signature Size	PK Size
[SSH11] (3 rounds)	—	28 502 B	38 B
MQ-DSS [CHR+16]	—	41 444 B	
MudFish [Beu20]	—	14 640 B	
Mesquite [Wan22]	Fast	9 578 B	
	Short	**8 609 B**	
Our scheme	Fast	10 764 B	
	Short	9 064 B	

$q = 256$
$m = 40$
$n = 40$

	Variant	Signature Size	PK Size
[SSH11] (3 rounds)	—	40 328 B	56 B
MQ-DSS [CHR+16]	—	28 768 B	
MudFish [Beu20]	Fast	15 958 B	
	Short	13 910 B	
Mesquite [Wan22]	Fast	11 339 B	
	Short	9 615 B	
Our scheme	Fast	8 488 B	
	Short	**7 114 B**	

MPC protocols for MinRank and Rank SD

MinRank Problem

From $(M_0, M_1, \ldots, M_k)$, find $x \in \mathbb{F}_q^k$ such that

$$\mathrm{rank}(M_0 + \sum_{i=1}^{k} x_i M_i) \leq r.$$

Rank Syndrome Decoding Problem

From (H, y), find $x \in \mathbb{F}_{q^m}^n$ such that

$$y = Hx \qquad \text{and} \qquad \mathrm{rank}(x) \leq r.$$

MPC protocols for MinRank and Rank SD

MinRank Problem

From $(M_0, M_1, \ldots, M_k)$, find $x \in \mathbb{F}_q^k$ such that

$$\mathrm{rank}\left(\boxed{M_0 + \sum_{i=1}^{k} x_i M_i}\right) \leq r.$$

Rank Syndrome Decoding Problem

From (H, y), find $x \in \mathbb{F}_{q^m}^n$ such that

$$\boxed{y = Hx} \quad \text{and} \quad \mathrm{rank}(x) \leq r.$$

Linear into the secret values

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r.

Rank Decomposition:

A matrix $M \in \mathbb{F}_q^{n \times m}$ has a rank of at most r
iff there exists $T \in \mathbb{F}_q^{n \times r}$ and $R \in \mathbb{F}_q^{r \times m}$ such that $M = TR$.

Inputs: M, T and R.

1. Check that $M = TR$

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r. Rewrite M as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

Linearized Polynomials:

A matrix $M \in \mathbb{F}_q^{n \times m}$ has a rank of at most r

$\Leftrightarrow$ there exists a linear subspace U of $\mathbb{F}_{q^m}$ of dimension r

such that $\{x_1, \ldots, x_n\} \subset U$

$\Leftrightarrow$ there exists a monic q-polynomial L_U of degree q^r

such that $x_1, \ldots, x_n$ are roots of L_U.

$$L_U := X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}$$

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $\boldsymbol{M} \in \mathbb{F}_q^{m \times n}$ has a rank of at most r. Rewrite $\boldsymbol{M}$ as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

Inputs: $\boldsymbol{M}$ and $L_U := X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}$.

We want to check that

$$L_U(x_1) = L_U(x_2) = \ldots = L_U(x_n) = 0.$$

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r. Rewrite M as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

Inputs: M and $L_U := X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}$.

We want to check that

$$0 = \sum_{j=1}^{n} \gamma_j \cdot L_U(x_j)$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $M \in \mathbb{F}_q^{m \times n}$ has a rank of at most r. Rewrite M as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

<u>Inputs</u>: M and $L_U := X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}$.

We want to check that

$$-\sum_{j=1}^{n} \gamma_j \cdot x_j^{q^r} = \langle \beta, \begin{pmatrix} \sum_{j=1}^n \gamma_j \cdot x_j^{q^0} \\ \vdots \\ \sum_{j=1}^n \gamma_j \cdot x_j^{q^{r-1}} \end{pmatrix} \rangle$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

MPC protocols for MinRank and Rank SD

The multi-party computation must check that the matrix $M \in \mathbb{F}_q^{m\times n}$ has a rank of at most r. Rewrite M as $(x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$.

Inputs: M and $L_U := X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}$.

Linear into the secret values

We want to check that

$$\boxed{-\sum_{j=1}^{n} \gamma_j \cdot x_j^{q^r}} = \left\langle \beta, \boxed{\begin{pmatrix} \sum_{j=1}^{n} \gamma_j \cdot x_j^{q^0} \\ \vdots \\ \sum_{j=1}^{n} \gamma_j \cdot x_j^{q^{r-1}} \end{pmatrix}} \right\rangle$$

where $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{ext}$ chosen by the verifier.

Signature Schemes from MinRank

$q = 16$
$m = 16$
$n = 16$
$k = 142$
$r = 4$

	Variant	*Signature Size*	*PK Size*
[Cou01]	—	28 575 B	73 B
[SINY22]	—	28 128 B	
[BESV22]	—	26 405 B	
[BG22]	Fast	13 644 B	
	Short	10 937 B	
[ARZV22]	Fast	10 116 B	
	Short	7 422 B	
Our scheme (rank decomposition)	Fast	9 288 B	
	Short	7 122 B	
Our scheme (q-polynomials)	Fast	7 204 B	
	Short	**5 518 B**	

Signature Schemes from RankSD

$q = 2$, $m = 31$, $n = 30$, $k = 15$, $r = 9$

	Variant	Signature Size	PK Size
[Ste94]	—	31 358 B	75 B
[Vér96]	—	27 115 B	
[FJR21]	—	19 328 B	
	—	14 181 B	
[BG22]	Fast	15 982 B	
	Short	12 274 B	
Our scheme (rank decomposition)	Fast	11 000 B	
	Short	8 543 B	
Our scheme (q-polynomials)	Fast	7 376 B	
	Short	**5 899 B**	

		Variant	Signature Size	PK Size
Ideal RSD	[BG22]	Fast	12 607 B	95 B
		Short	10 126 B	
Ideal RSL	[BG22]	Fast	9 392 B	410 B
		Short	6 754 B	

Conclusion

- Many ideas used in the current NIST candidates:

Fen24-MQ $\xrightarrow{\text{More advanced innerproduct checking}}$ MQOM

Fen24-RankSD (q-polynomials) $\xrightarrow[\text{+ small optimisation}]{}$ RYDE

Fen24-MinRank (q-polynomials) $\longrightarrow$ MIRA

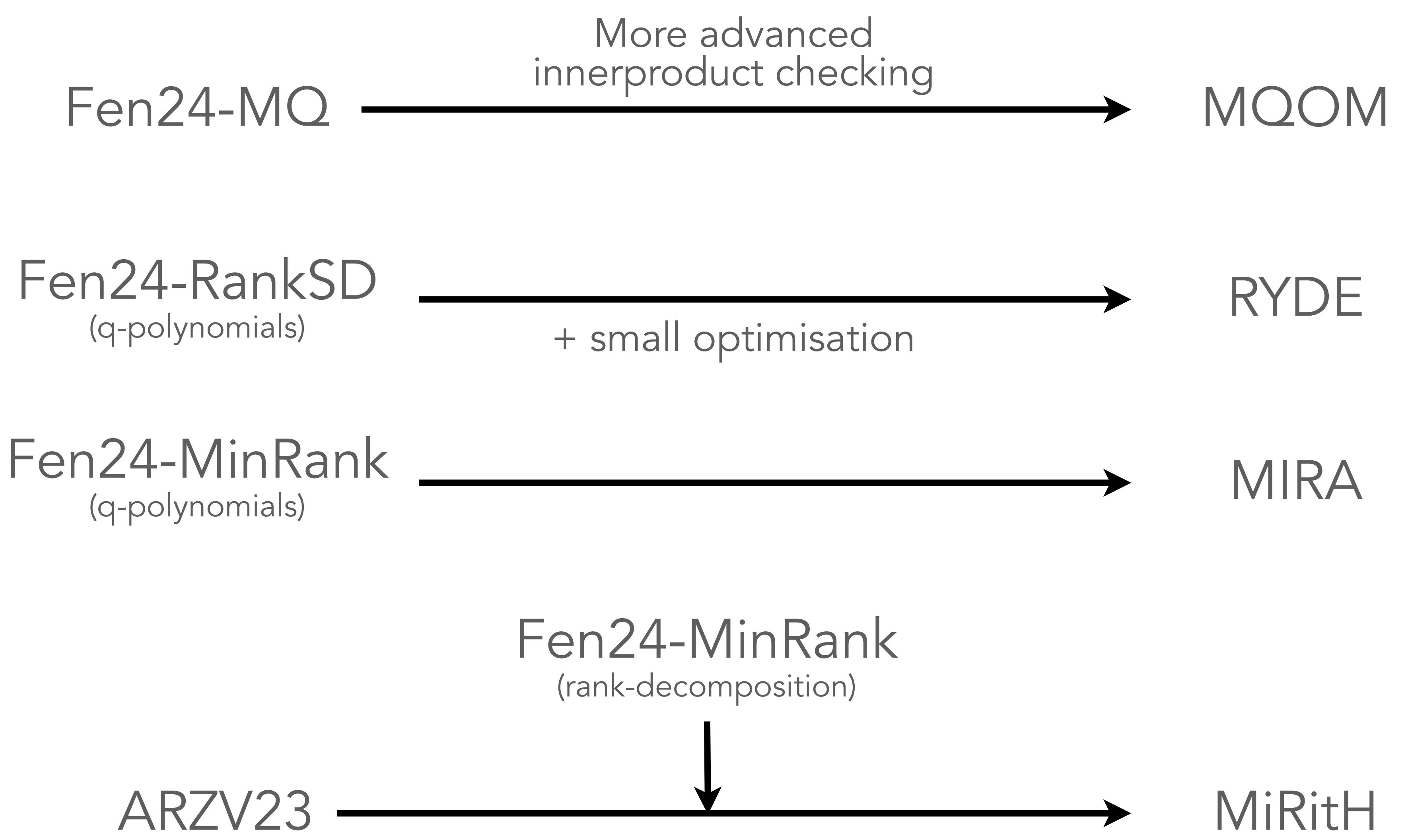
Fen24-MinRank (rank-decomposition)

ARZV23 $\longrightarrow$ MiRitH

[ARZV23] Adj, Rivera-Zamarripa,Verbel: "MinRank in the Head: Short Signatures from Zero-Knowledge Proofs" (AfricaCrypt 2023)

Conclusion

- Many ideas used in the current NIST candidates:



Thank you for your attention !