Post-Quantum Signatures from Secure Multiparty Computation

Thibauld Feneuil

Journées C2 - 2023

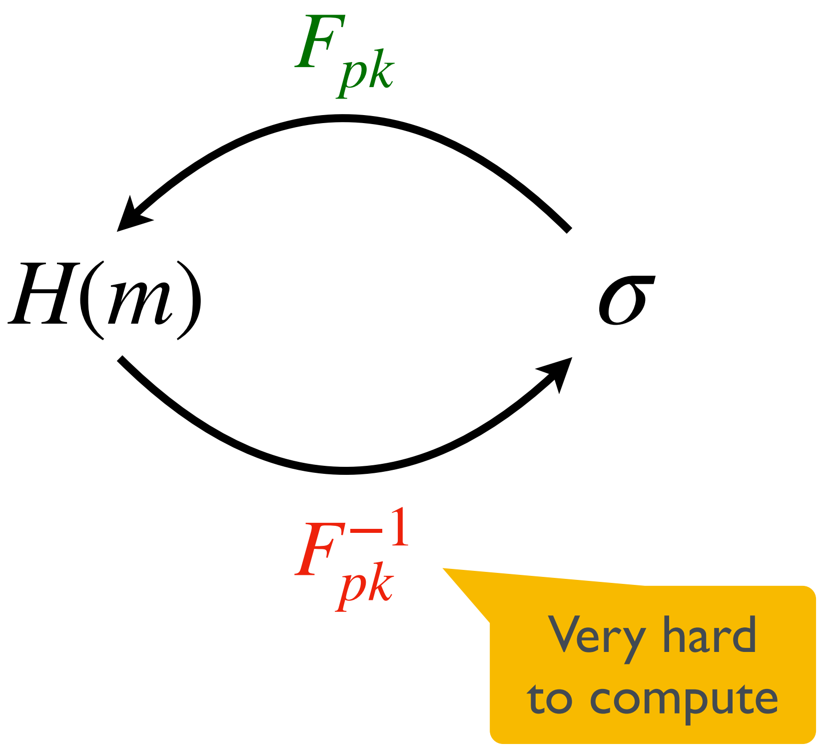
October 16, 2023, Najac (France)

CryptoExperts
WE INNOVATE TO SECURE YOUR BUSINESS
SORBONNE UNIVERSITÉ

Introduction

How to build signature schemes?

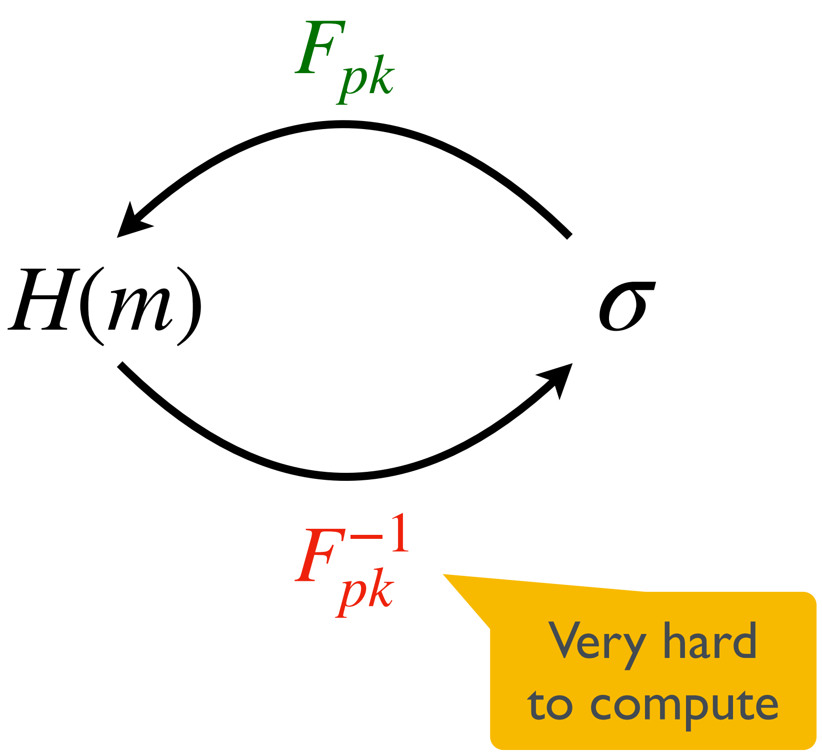
Hash & Sign



- Short signatures
- "Trapdoor" in the public key

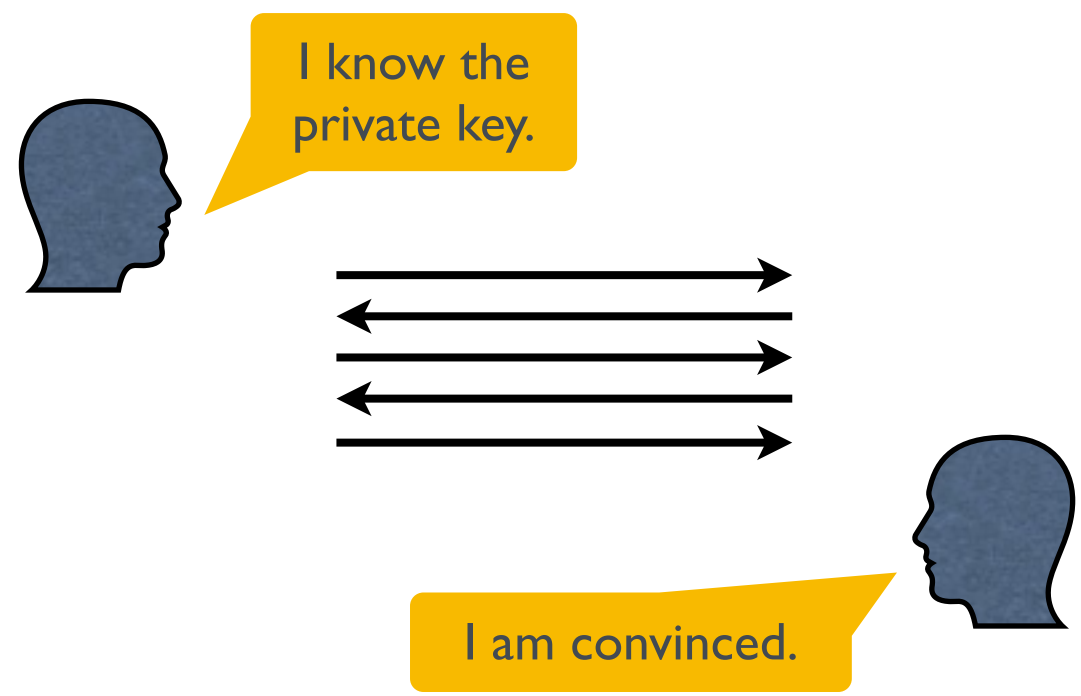
How to build signature schemes?

Hash & Sign



- Short signatures
- “Trapdoor” in the public key

From a zero-knowledge proof



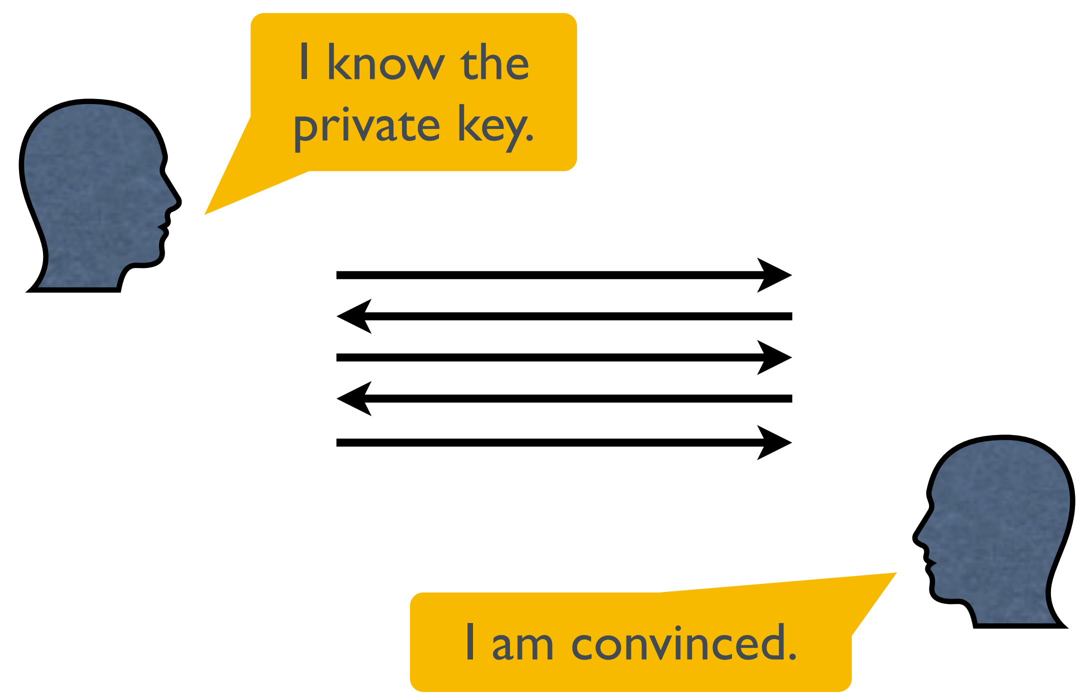
- Large(r) signatures
- Short public key

How to build signature schemes?

Hash & Sign

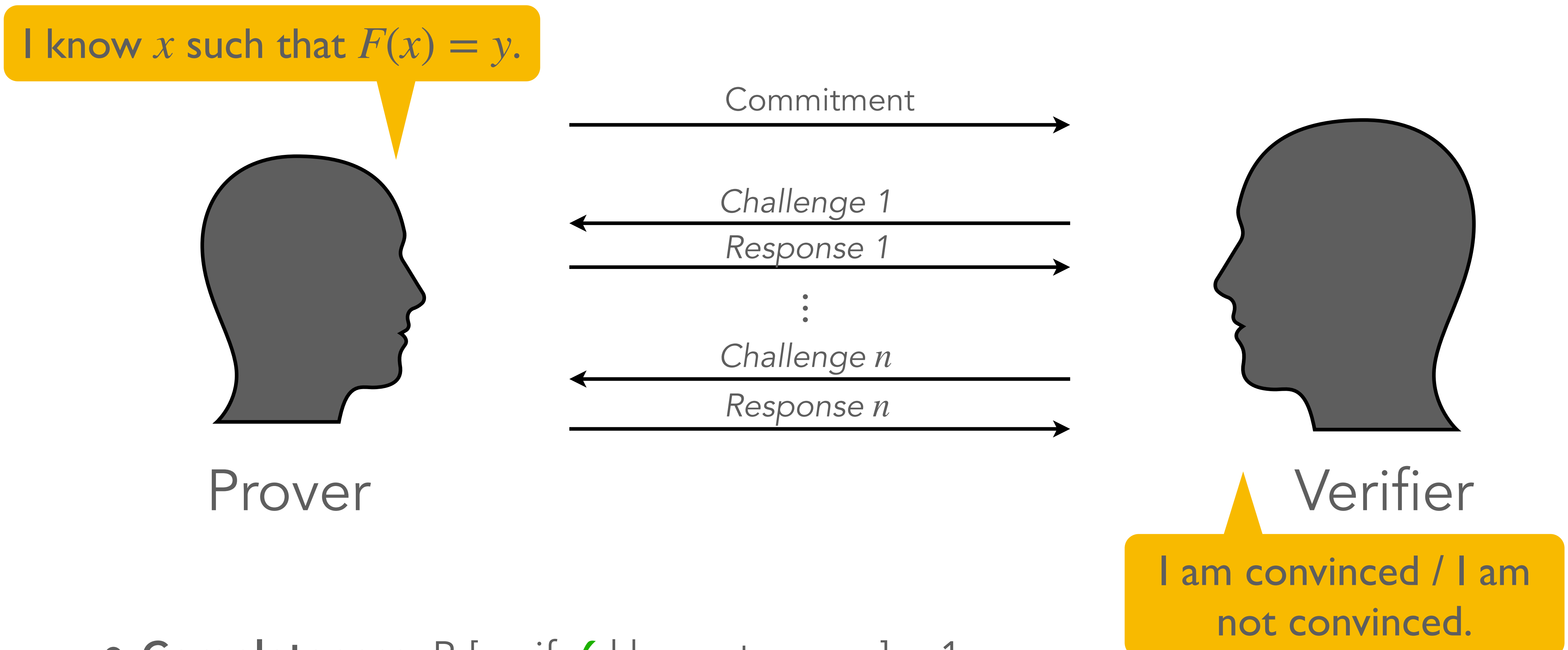
- Short signatures
- "Trapdoor" in the public key

From a zero-knowledge proof



- Large(r) signatures
- Short public key

Proof of knowledge

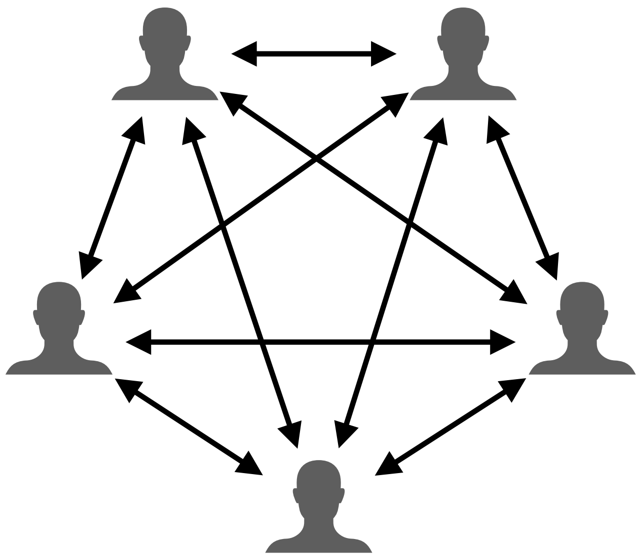


- **Completeness:** Pr[verif ✓ | honest prover] = 1
- **Soundness:** Pr[verif ✓ | malicious prover] $\leq \varepsilon$ (e.g. 2^{-128})
- **Zero-knowledge:** verifier learns nothing on x

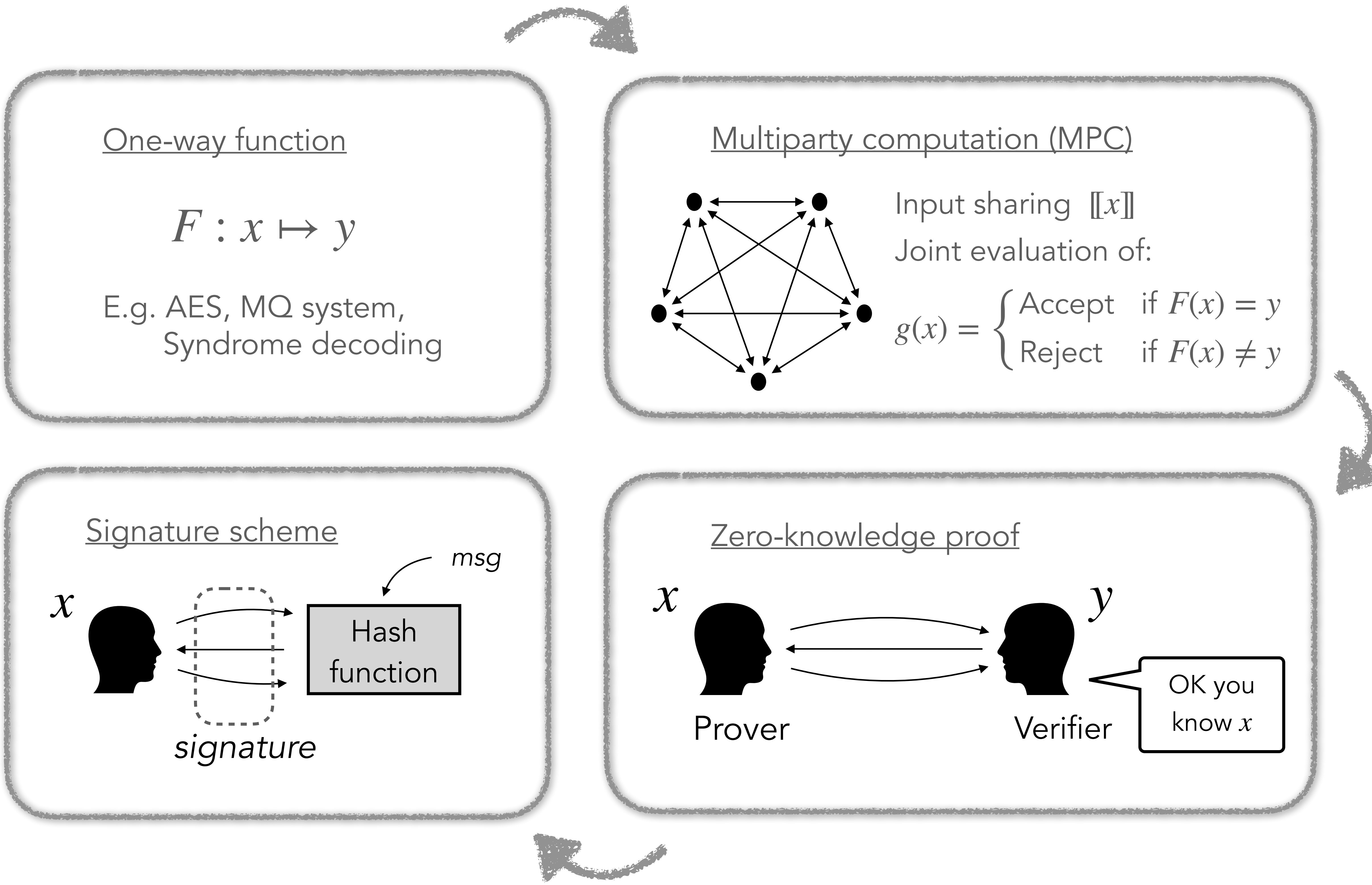
MPC in the Head

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

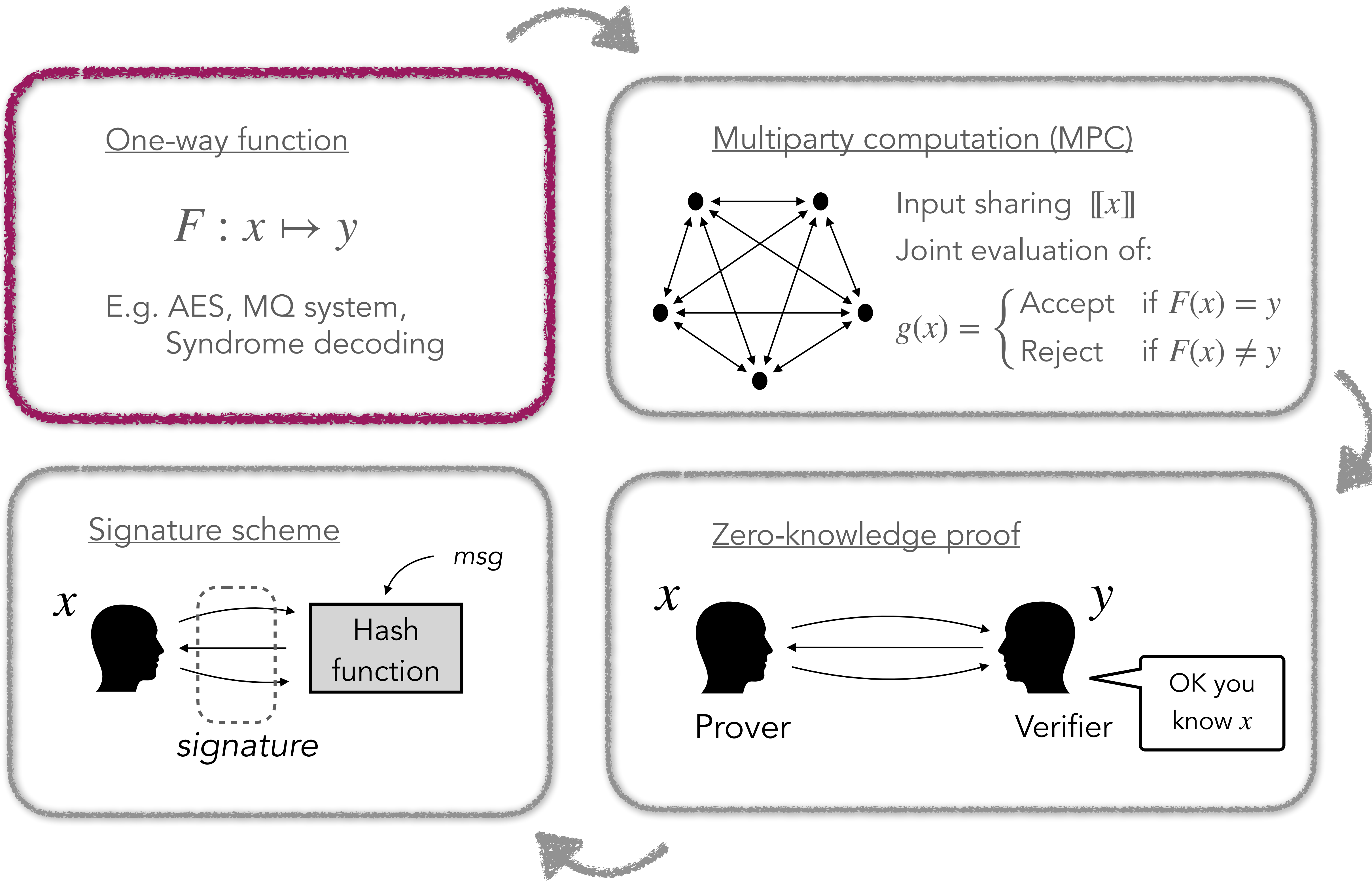
- Turn a *multiparty computation* (MPC) into an identification scheme / zero knowledge proof of knowledge

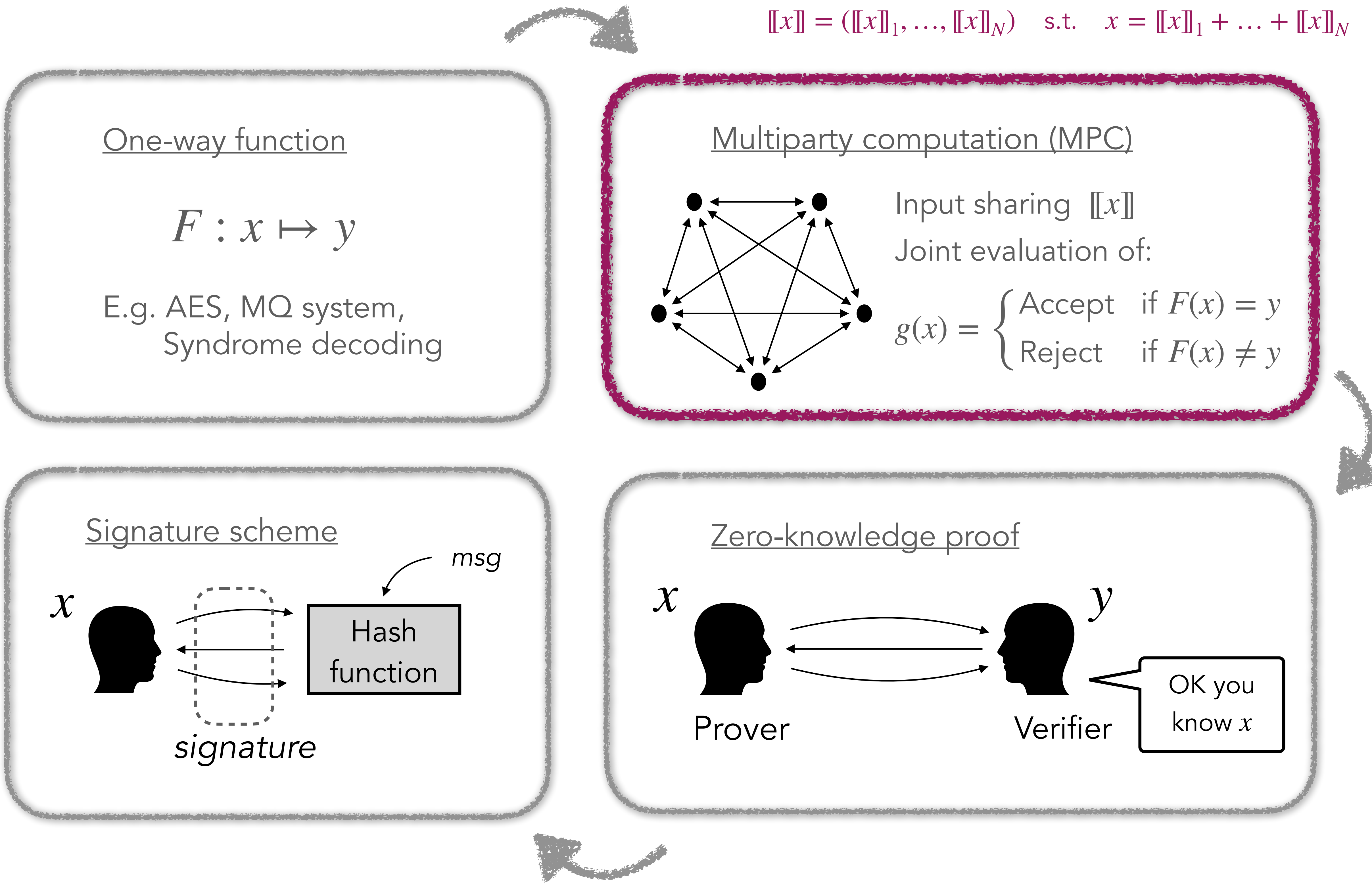


- **Generic**: can be apply to any cryptographic problem

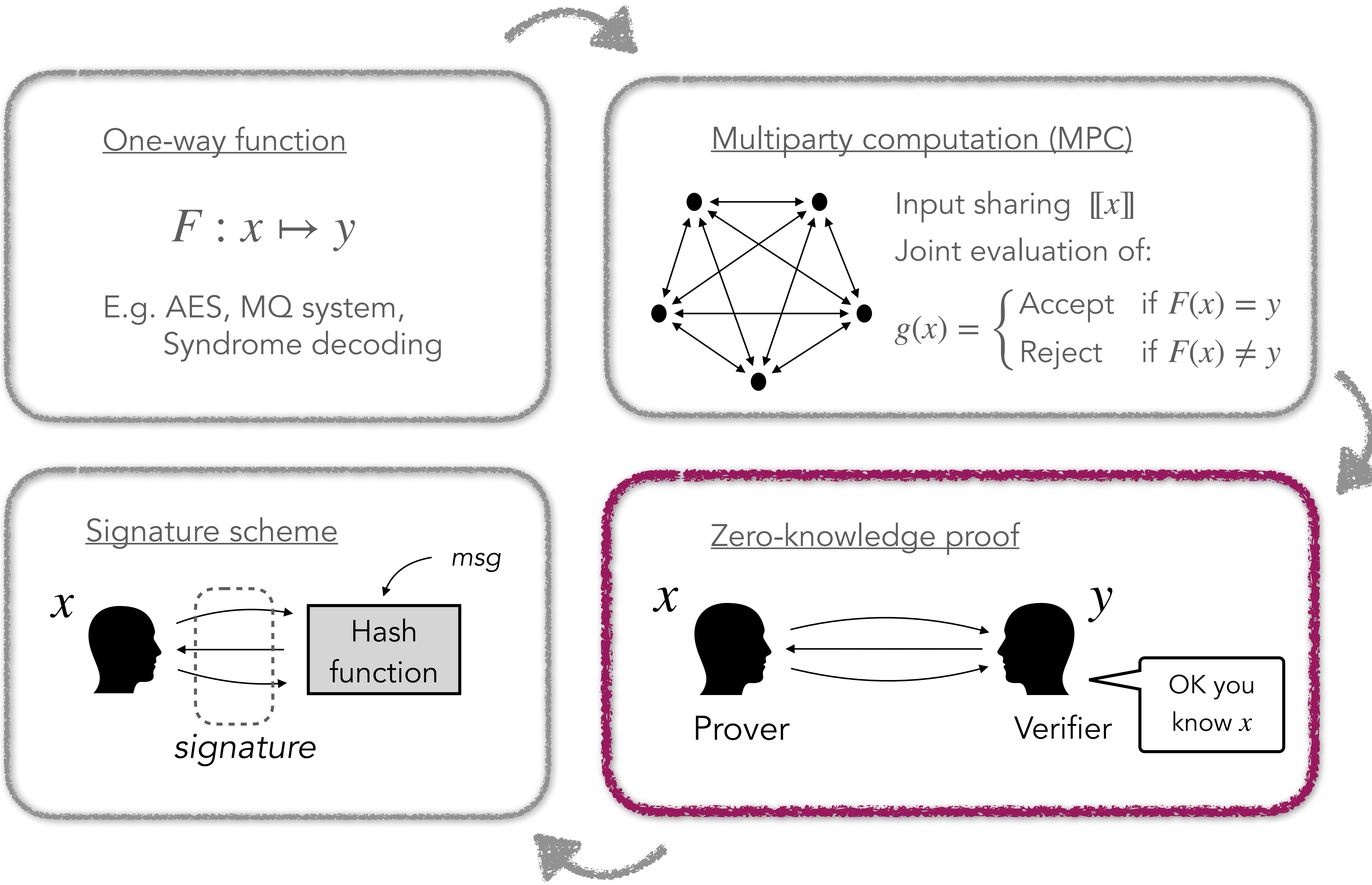


One-way function
$F : x \mapsto y$
E.g. AES, MQ system, Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Zero-knowledge proof
x
y
Prover
Verifier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

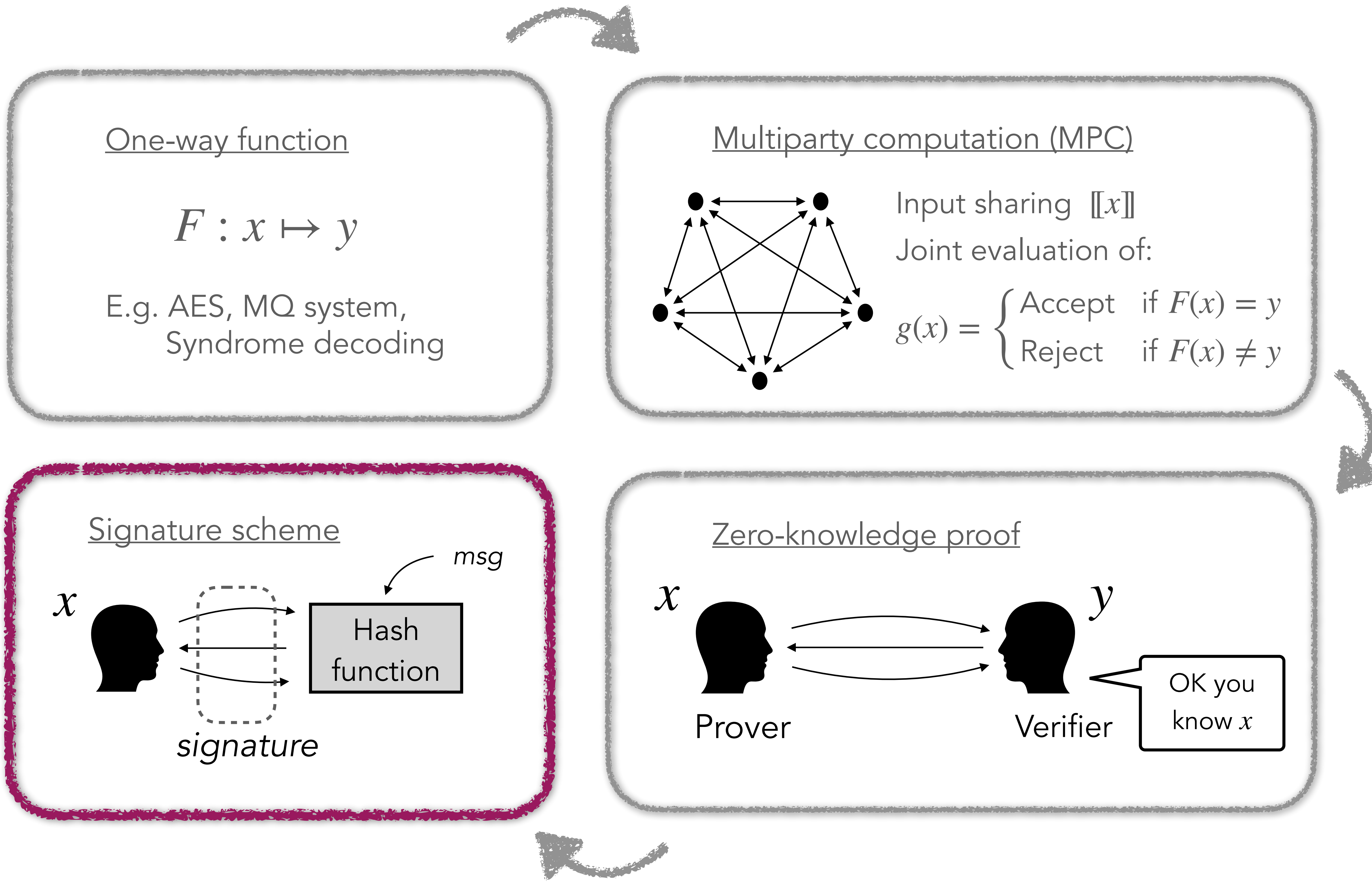

One-way function
$F : x \mapsto y$
E.g. AES, MQ system, Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Signature scheme
x
msg
Hash function
signature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



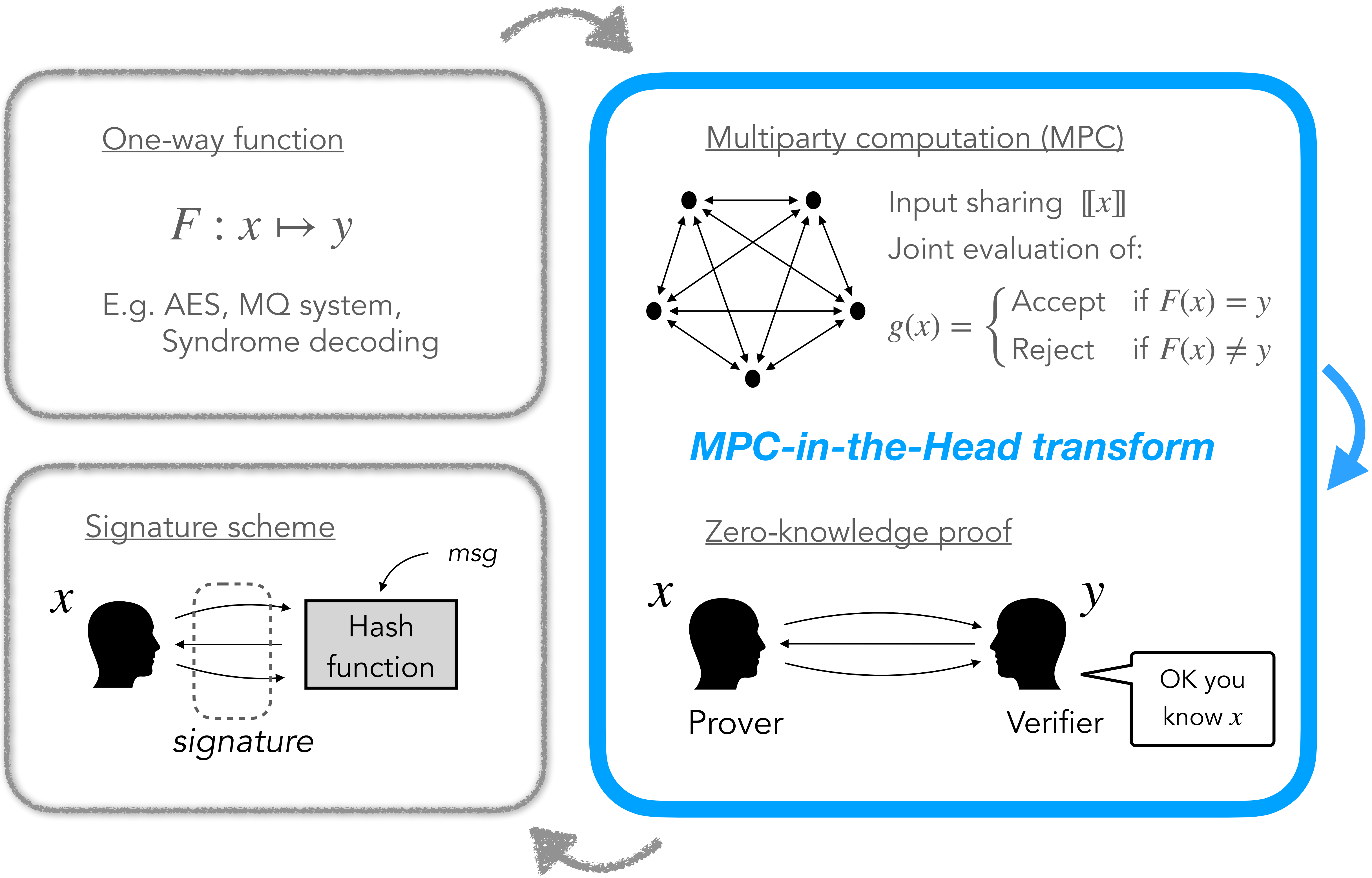
$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$ s.t. $x = [\![x]\!]_1 + \ldots + [\![x]\!]_N$
One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
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
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
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

MPCitH transform

Prover

Verifier

MPCitH transform

① Generate and commit shares

$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$

$\cdots$

$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$

Prover

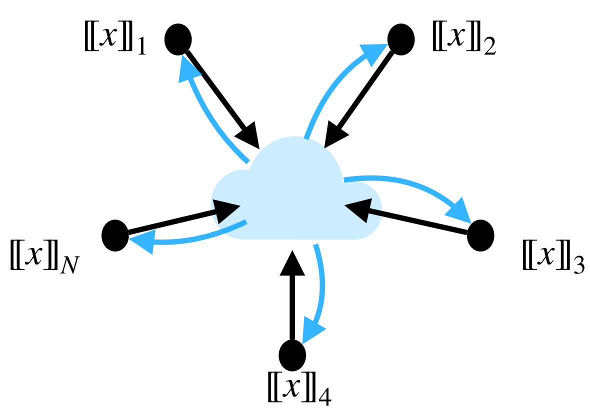
Verifier

MPCitH transform

① Generate and commit shares

$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

② Run MPC in their head



$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$

...

$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$

send broadcast

$[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$

Prover

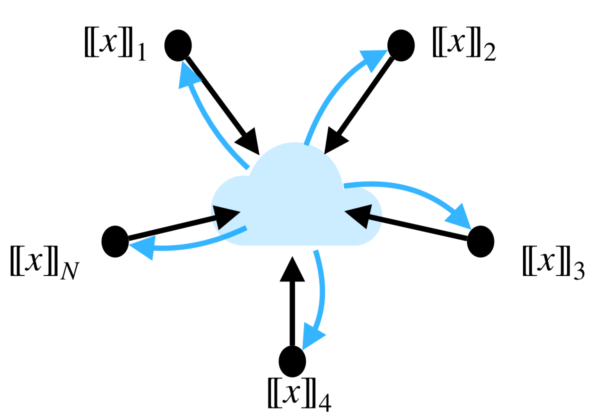
Verifier

MPCitH transform

① Generate and commit shares

$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

② Run MPC in their head



$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$
…
$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$

send broadcast
$[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$

i^*

③ Choose a random party

$i^* \xleftarrow{\$} \{1,\ldots,N\}$

Prover

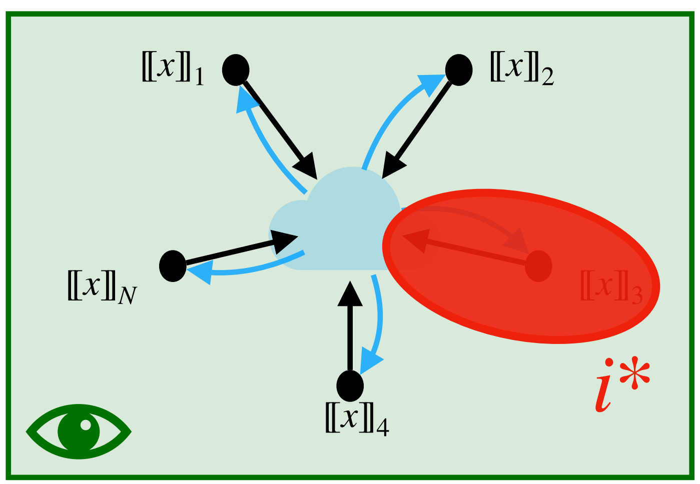
Verifier

MPCitH transform

① Generate and commit shares

$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

② Run MPC in their head



④ Open parties $\{1,\ldots,N\} \backslash \{i^*\}$

$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$

$\cdots$

$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$ →

send broadcast

$[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$ →

← i^*

$([\![x]\!]_i, \rho_i)_{i \neq i^*}$ →

③ Choose a random party

$i^* \xleftarrow{\$} \{1,\ldots,N\}$

Prover

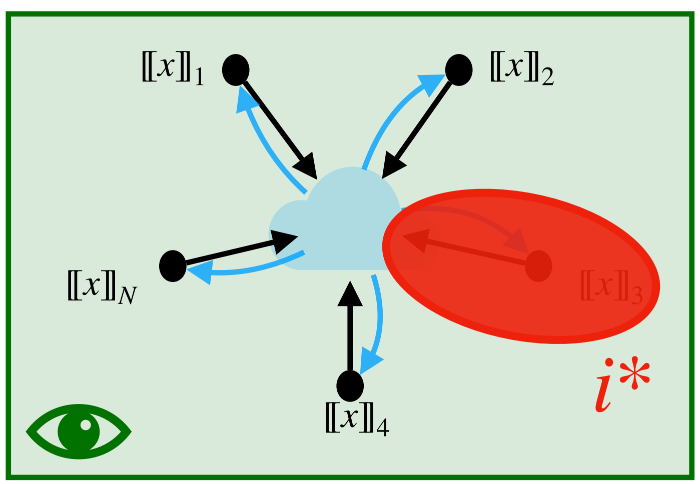
Verifier

MPCitH transform

① Generate and commit shares

$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

② Run MPC in their head



④ Open parties $\{1,\ldots,N\}\backslash\{i^*\}$

$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$

$\ldots$

$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$ →

send broadcast

$[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$ →

← i^*

$([\![x]\!]_i, \rho_i)_{i \neq i^*}$ →

③ Choose a random party

$i^* \leftarrow^{\$} \{1,\ldots,N\}$

⑤ Check $\forall i \neq i^*$

- Commitments $\mathrm{Com}^{\rho_i}([\![x]\!]_i)$
- MPC computation $[\![\alpha]\!]_i = \varphi([\![x]\!]_i)$

Check $g(y, \alpha) =$ Accept

Prover

Verifier

MPCitH transform

① Generate and commit shares

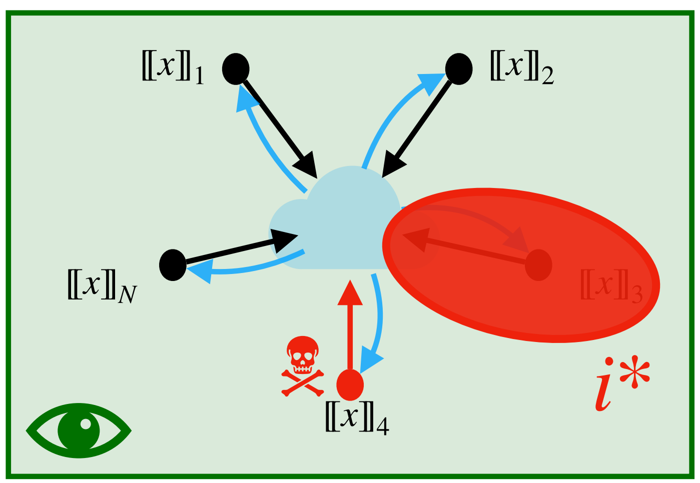
$[\![x]\!] = ([\![x]\!]_1, \ldots, [\![x]\!]_N)$

We have $F(x) \neq y$ where

$x := [\![x]\!]_1 + \ldots + [\![x]\!]_N$

$\mathrm{Com}^{\rho_1}([\![x]\!]_1)$
…
$\mathrm{Com}^{\rho_N}([\![x]\!]_N)$ →

② Run MPC in their head



send broadcast
$[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$ →

③ Choose a random party

$i^* \xleftarrow{\$} \{1, \ldots, N\}$

← i^*

④ Open parties $\{1, \ldots, N\} \backslash \{i^*\}$

$([\![x]\!]_i, \rho_i)_{i \neq i^*}$ →

⑤ Check $\forall i \neq i^*$

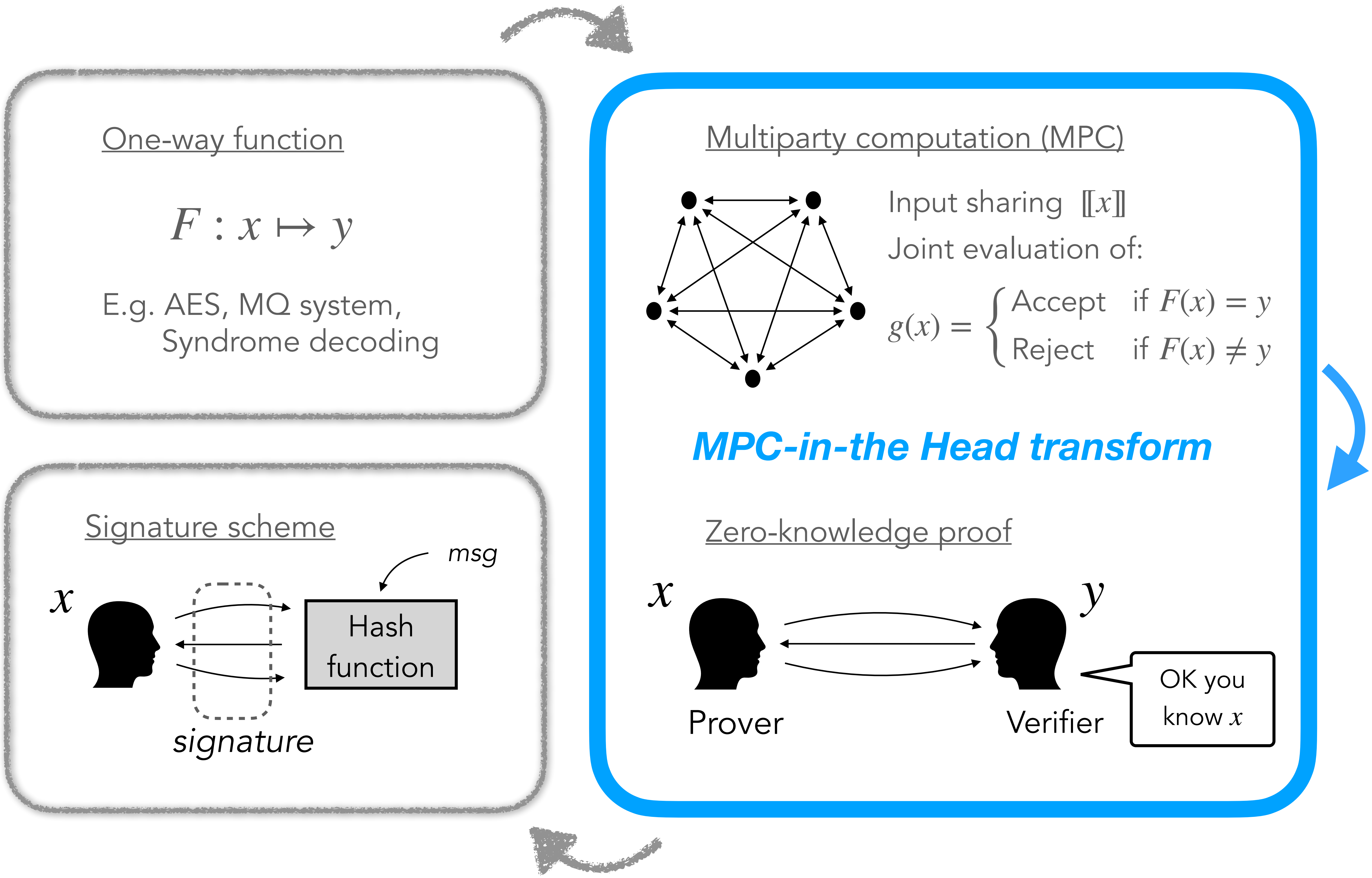
- Commitments $\mathrm{Com}^{\rho_i}([\![x]\!]_i)$
- MPC computation $[\![\alpha]\!]_i = \varphi([\![x]\!]_i)$

Check $g(y, \alpha) =$ Accept

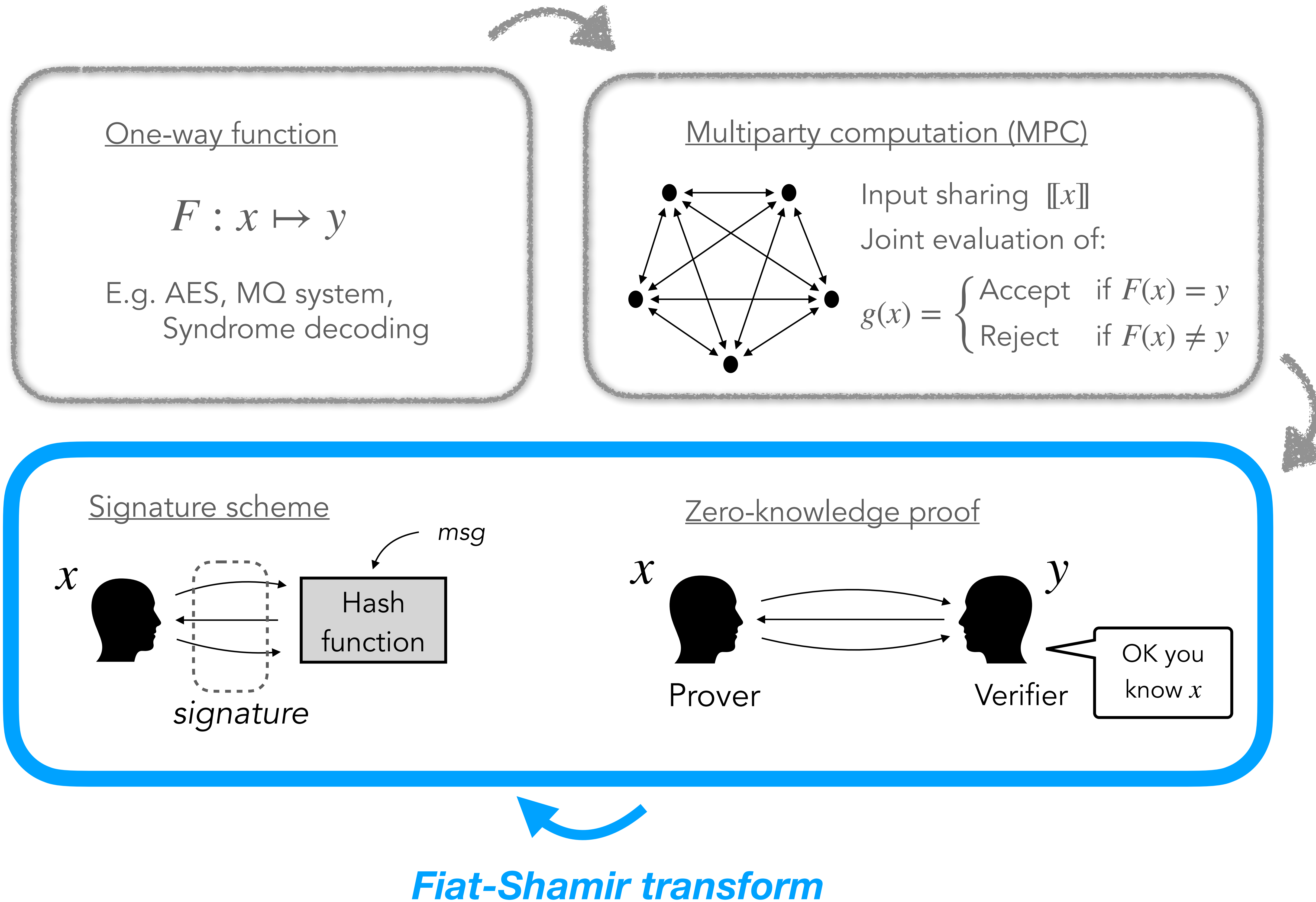
Malicious Prover

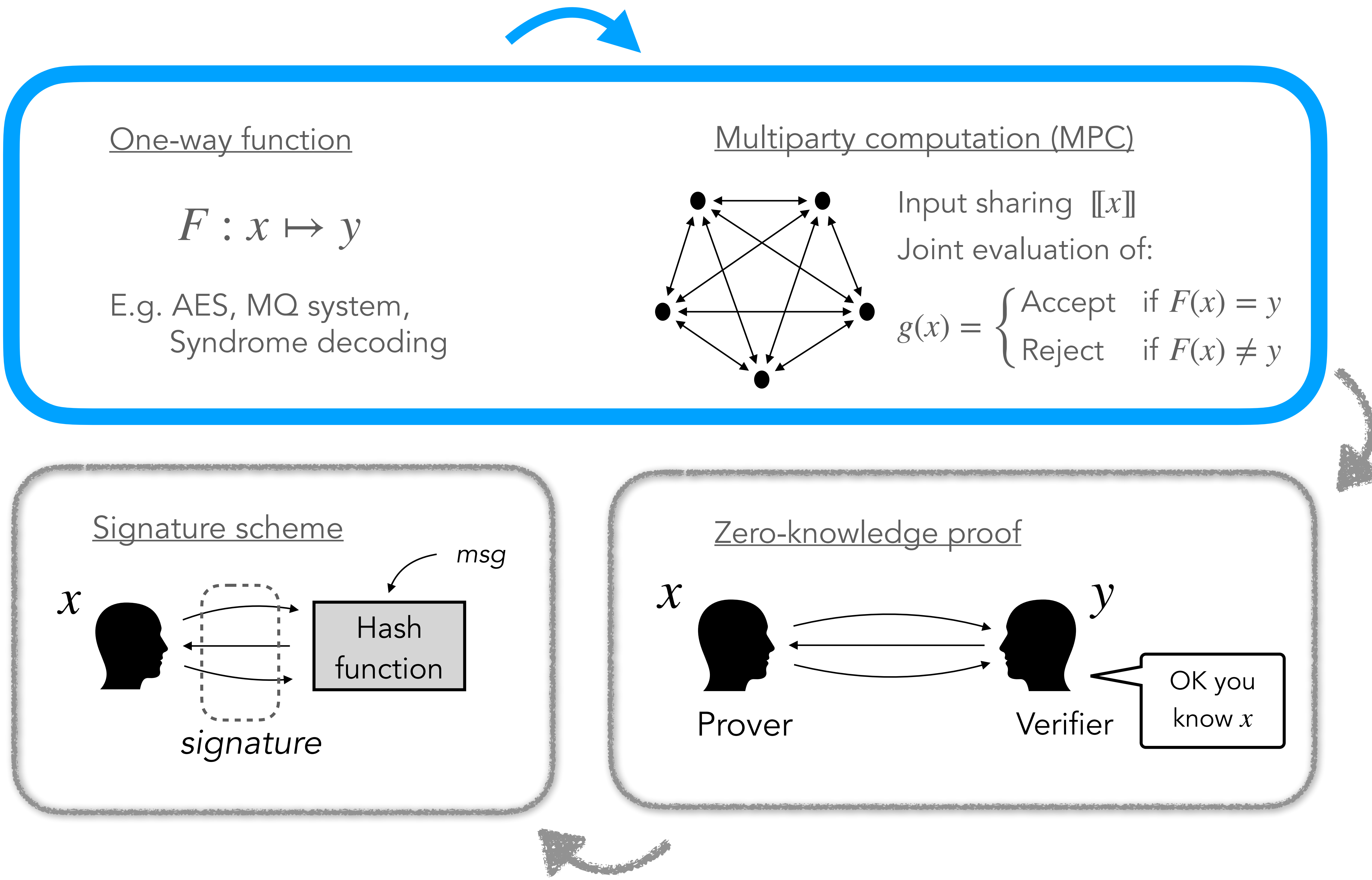
Verifier

Cheating detected!

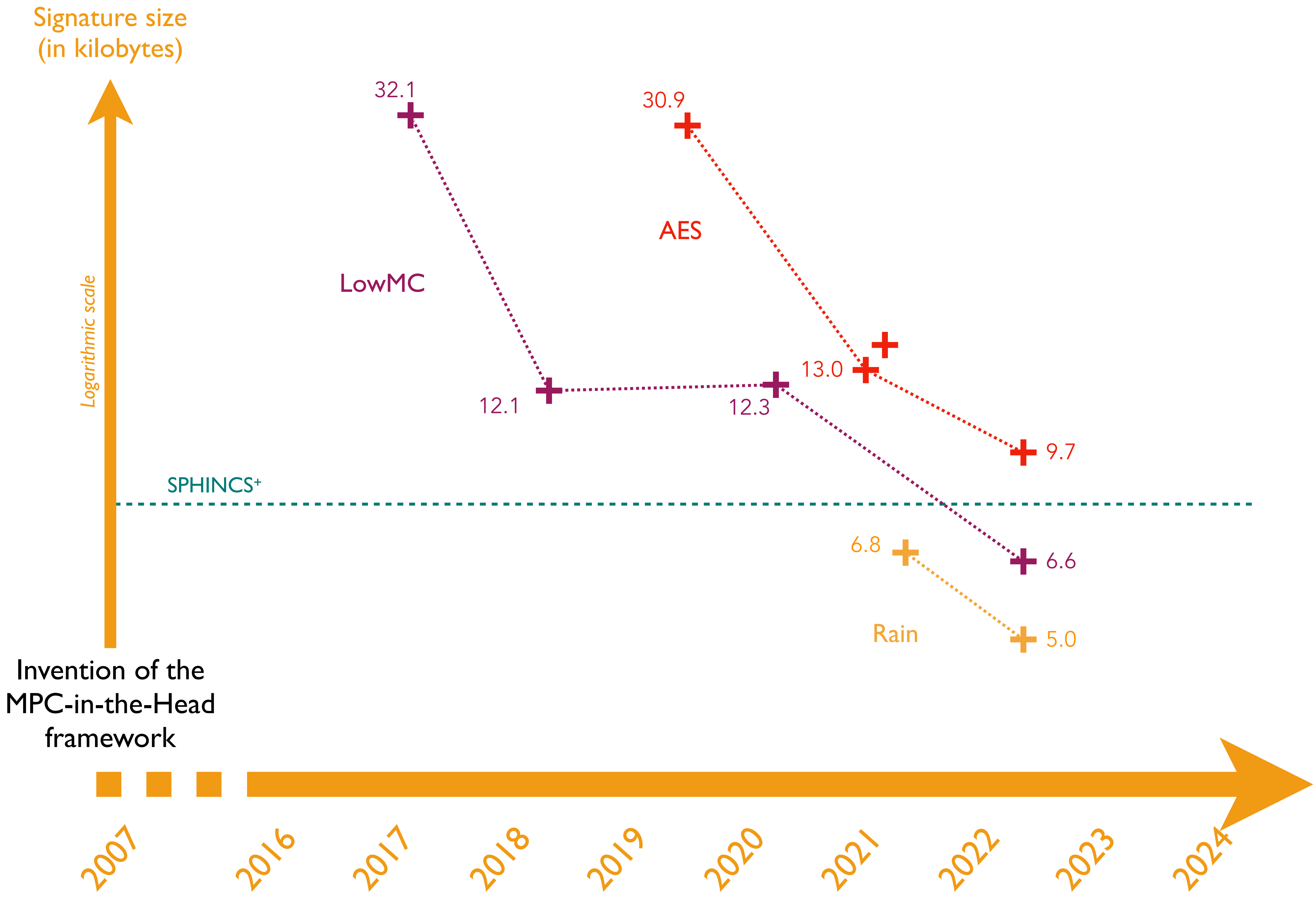


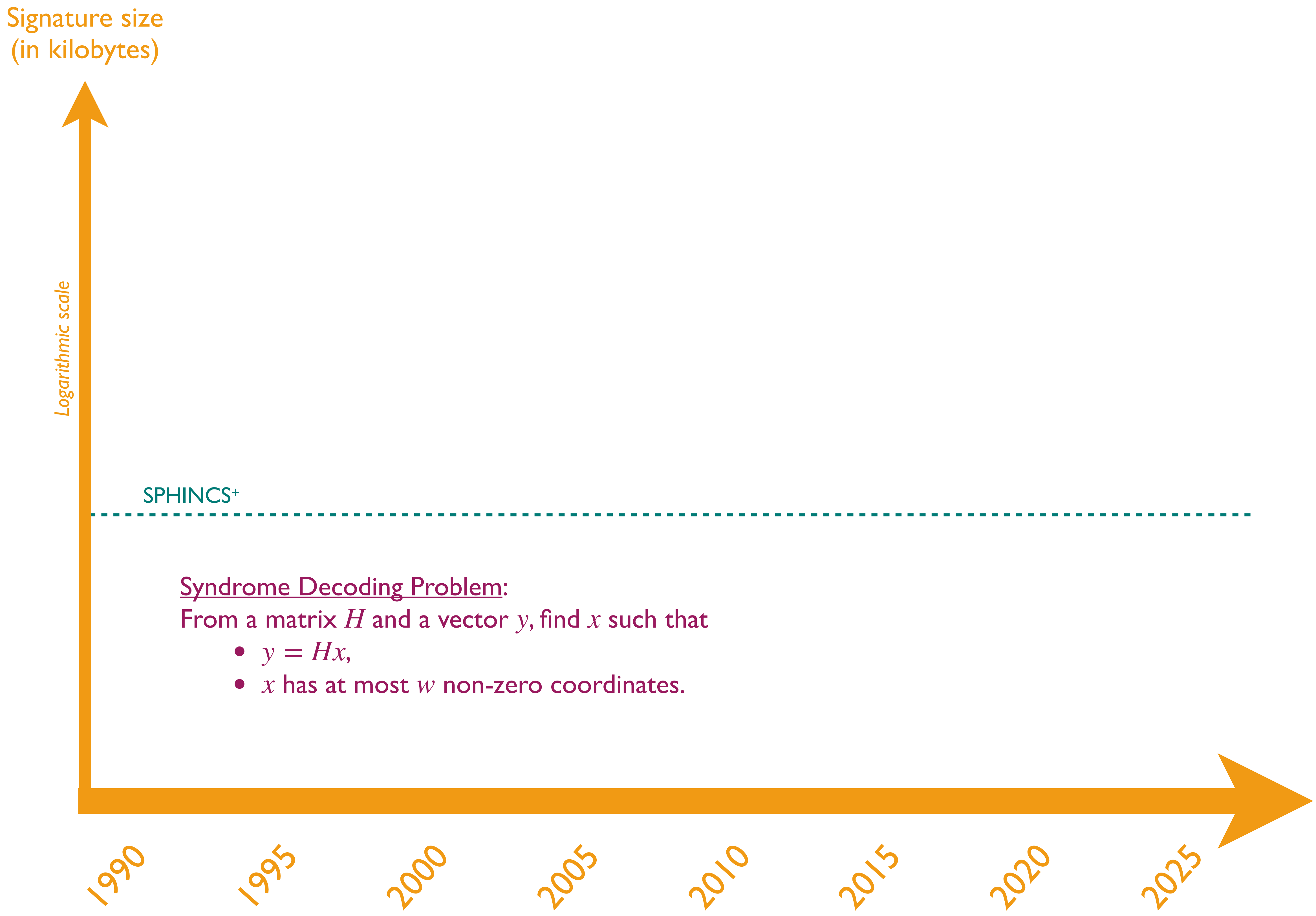
One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Signature scheme
x
msg
Hash function
signature
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$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



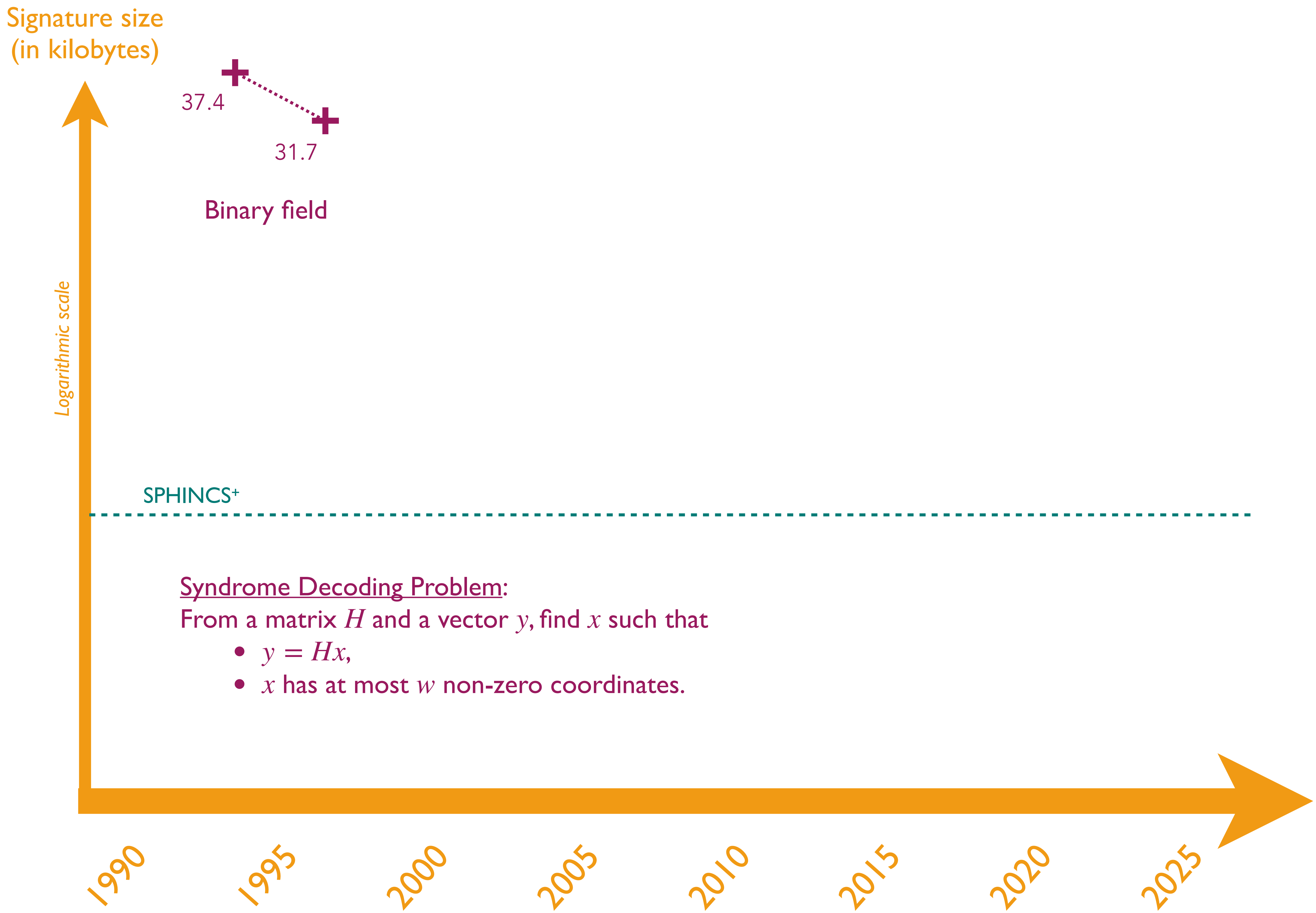
One-way function
$F : x \mapsto y$
E.g. AES, MQ system,
Syndrome decoding
Multiparty computation (MPC)
Input sharing $[\![x]\!]$
Joint evaluation of:
$g(x) = \begin{cases} \text{Accept} & \text{if } F(x) = y \\ \text{Reject} & \text{if } F(x) \neq y \end{cases}$
Signature scheme
x
msg
Hash function
signature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


Signature size
(in kilobytes)
Logarithmic scale
32.1
30.9
AES
LowMC
13.0
12.1
12.3
9.7
SPHINCS+
6.8
6.6
Rain
5.0
Invention of the
MPC-in-the-Head
framework
2007
2016
2017
2018
2019
2020
2021
2022
2023
2024



Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

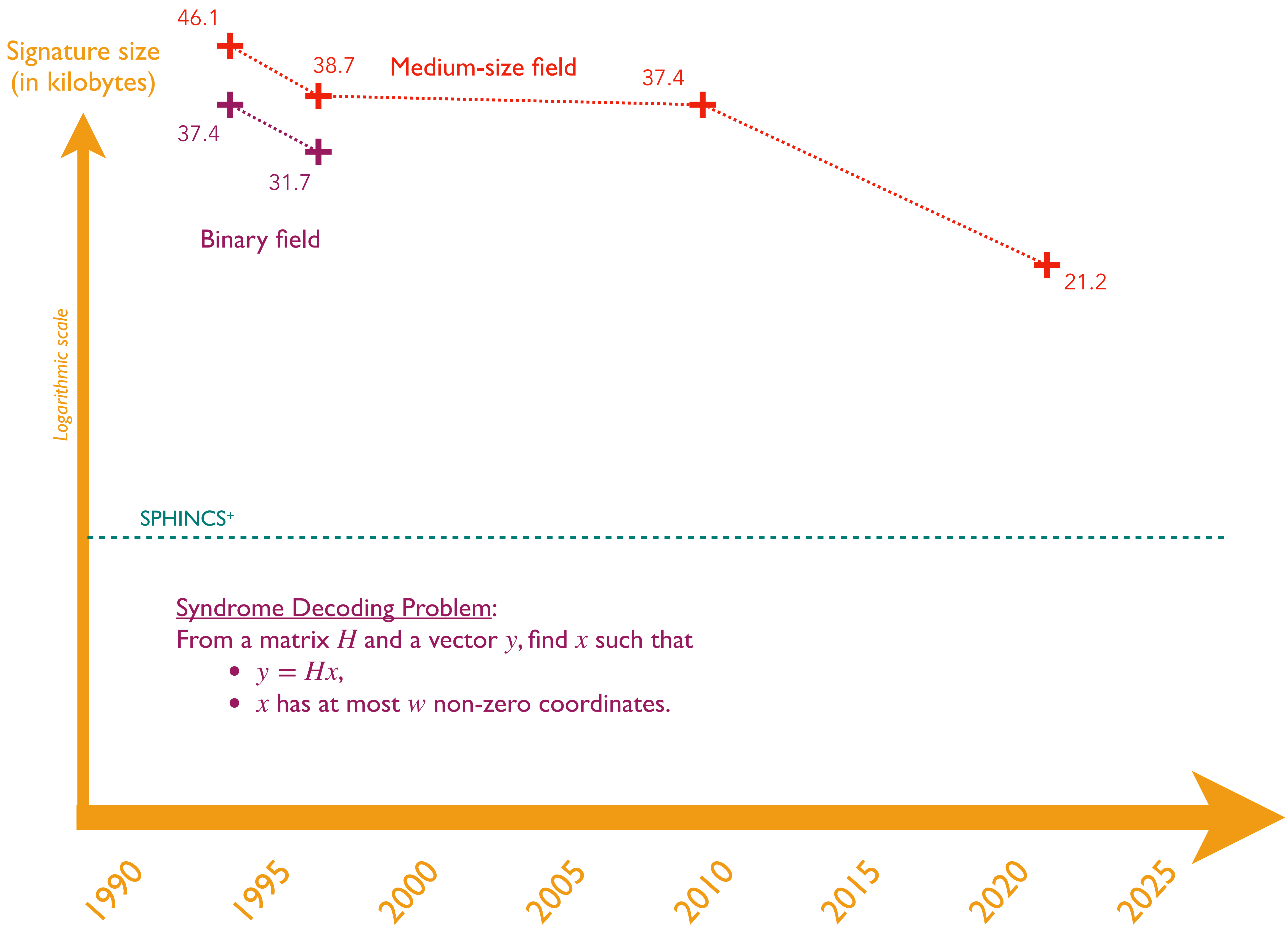
- $y = Hx$,
- x has at most w non-zero coordinates.



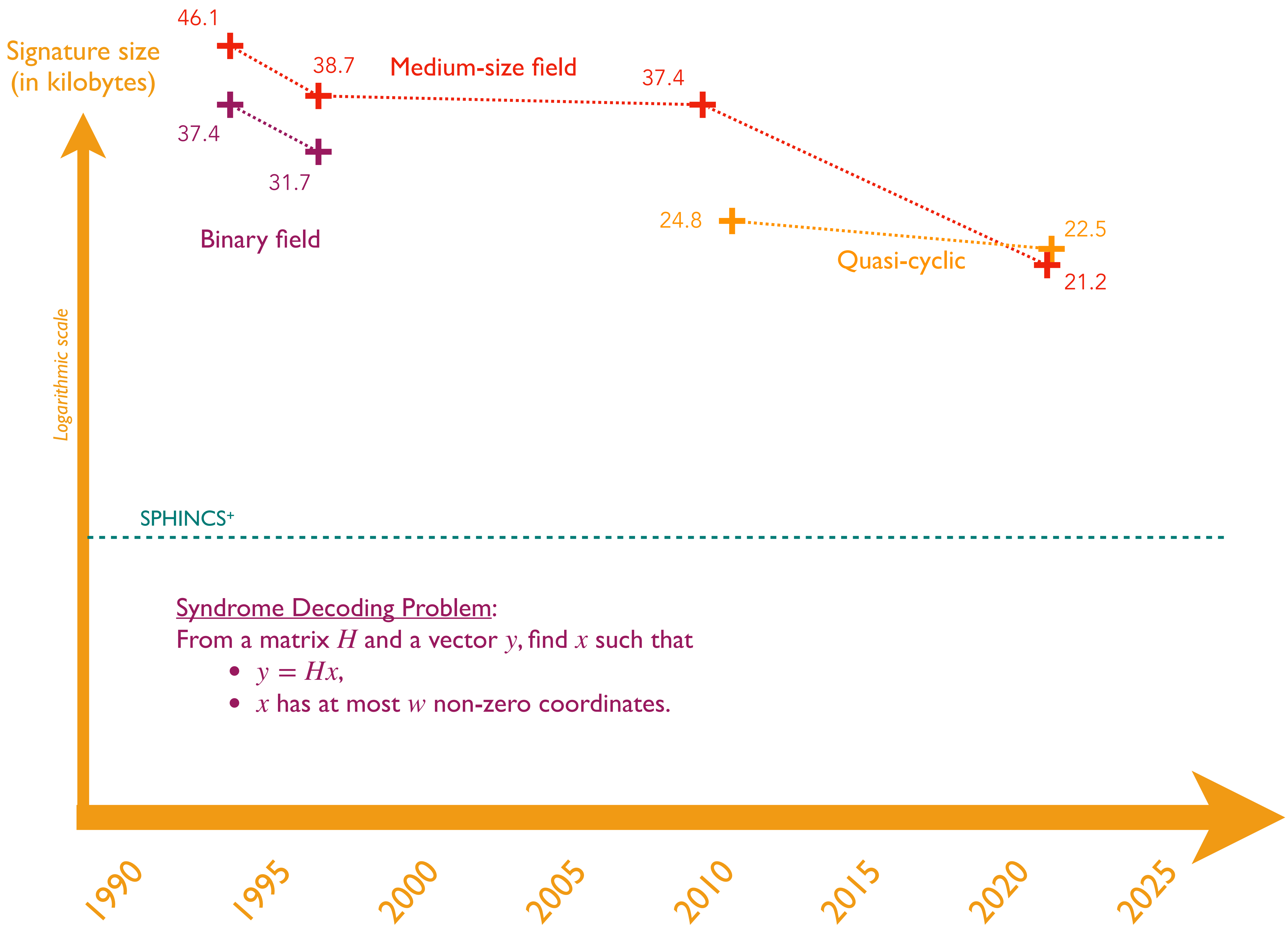
<u>Syndrome Decoding Problem</u>:

From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

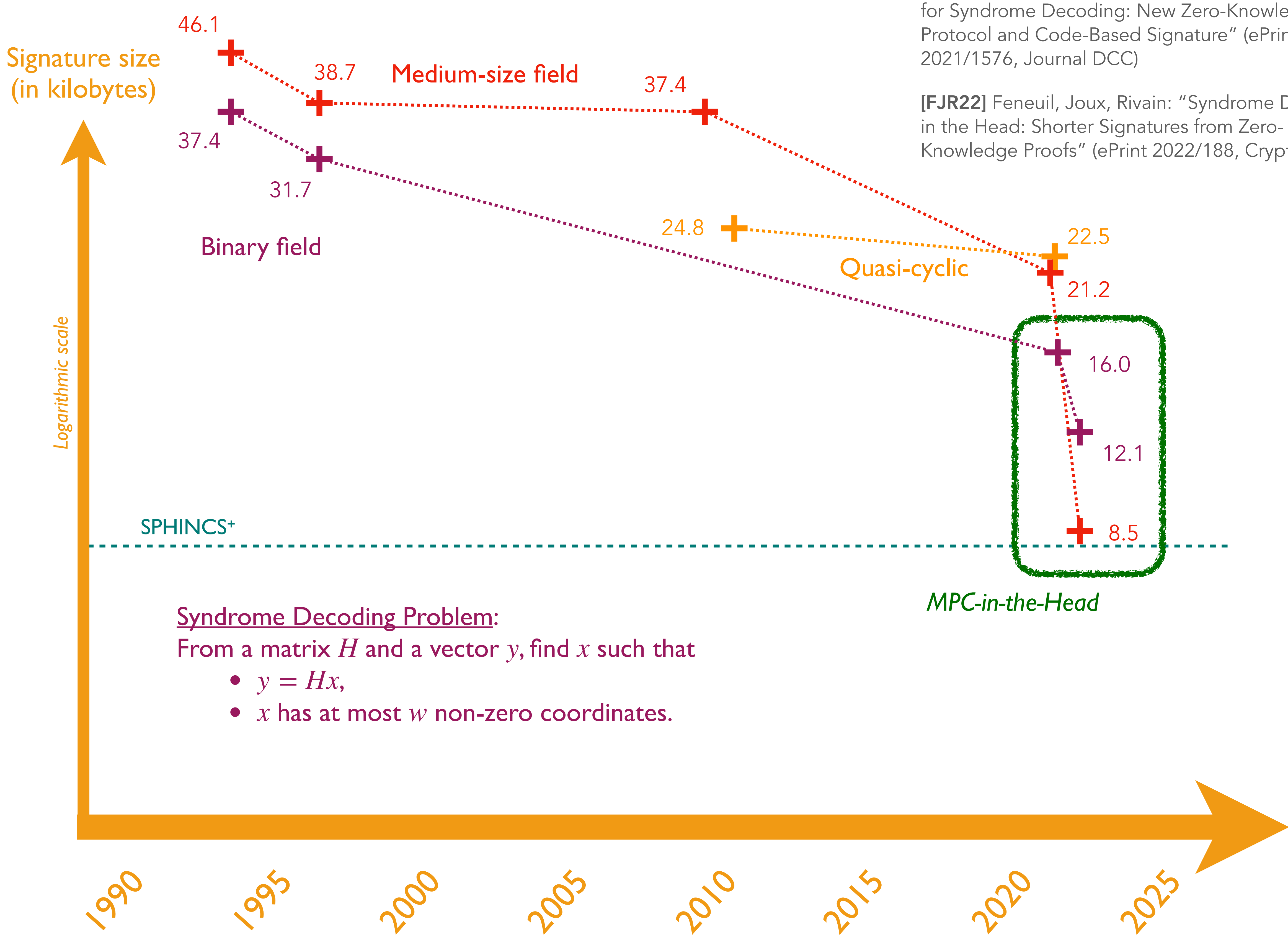


Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that
- $y = Hx$,
- x has at most w non-zero coordinates.



Syndrome Decoding Problem:
From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.



[FJR23] Feneuil, Joux, Rivain: "Shared Permutation for Syndrome Decoding: New Zero-Knowledge Protocol and Code-Based Signature" (ePrint 2021/1576, Journal DCC)

[FJR22] Feneuil, Joux, Rivain: "Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs" (ePrint 2022/188, Crypto 2022)

Syndrome Decoding Problem:

From a matrix H and a vector y, find x such that

- $y = Hx$,
- x has at most w non-zero coordinates.

Exploring other assumptions

- **Subset Sum Problem:** ≥ 100 KB $\Rightarrow$ 19.1 KB
- **Multivariate Quadratic Problem:** 6.3 – 7.3 KB
- **MinRank Problem:** $\approx 5 - 6$ KB
- **Rank Syndrome Decoding Problem:** $\approx 5 - 6$ KB
- **Permuted Kernel Problem (or variant):** ≈ 6 KB
- …

MPCitH-based NIST Candidates

1st June 2023:

Deadline for the NIST call
for additional post-quantum signatures

MPCitH-based NIST Candidates

	Assumption	Size (in KB)
AIMer	AIM (MPC-friendly one-way function)	4.2
Biscuit	Structured MQ problem (PowAff2)	4.7
MIRA	MinRank problem	5.6
MiRitH	MinRank problem	5.7
RYDE	Syndrome decoding problem in rank metric	6.0
PERK*	Permuted Kernel problem (variant)	6.1
MQOM	Unstructured MQ problem	6.3
SDitH	Syndrome decoding problem in Hamming	8.2

MPCitH-based NIST Candidates

	Assumption	Size (in KB)
AIMer	AIM (MPC-friendly one-way function)	4.2
Biscuit	Structured MQ problem (PowAff2)	4.7
MIRA	MinRank problem	5.6
MiRitH	MinRank problem	5.7
RYDE	Syndrome decoding problem in rank metric	6.0
PERK	Permuted Kernel problem (variant)	6.1
MQOM	Unstructured MQ problem	6.3
SDitH	Syndrome decoding problem in Hamming	8.2

What about the computational cost ?

Traditional Transformation

(2018) Emulation : N parties

Traditional Transformation (2018)

Emulation : N parties

Hypercube Technique (2022)

$1 + \log_2 N$ parties

No communication penalty

[AGHHJY23] Aguilar-Melchor, Gama, Howe, Hülsing, Joseph, Yue: "The Return of the SDitH" (ePrint 2022/1645, Eurocrypt 2023)

Traditional Transformation

(2018)

Emulation : N parties

Hypercube Technique

(2022)

$1 + \log_2 N$ parties

No communication penalty

Threshold Approach

(2022)

Shamir's secret sharings

$1 + \ell$ parties

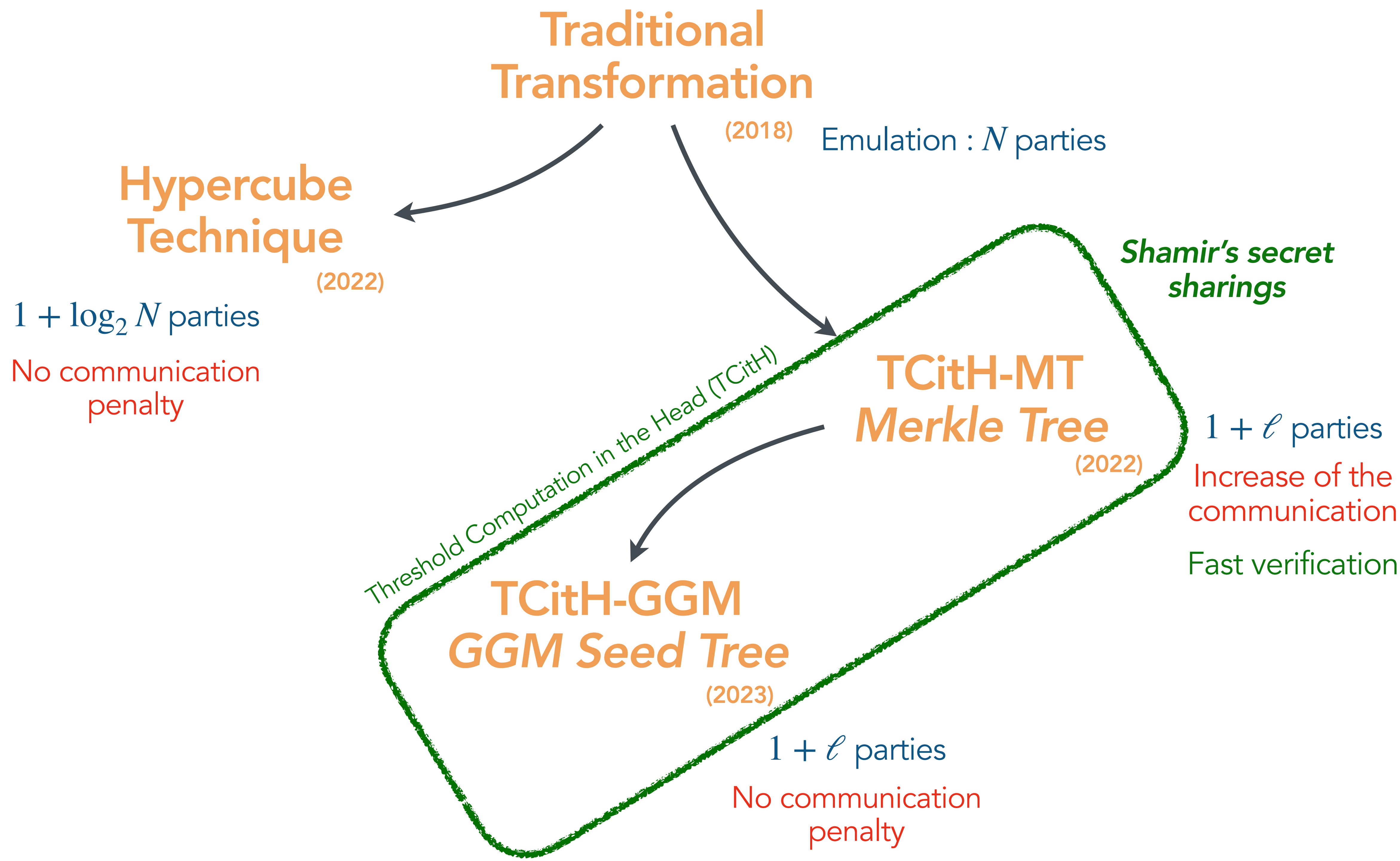
Increase of the communication

Fast verification

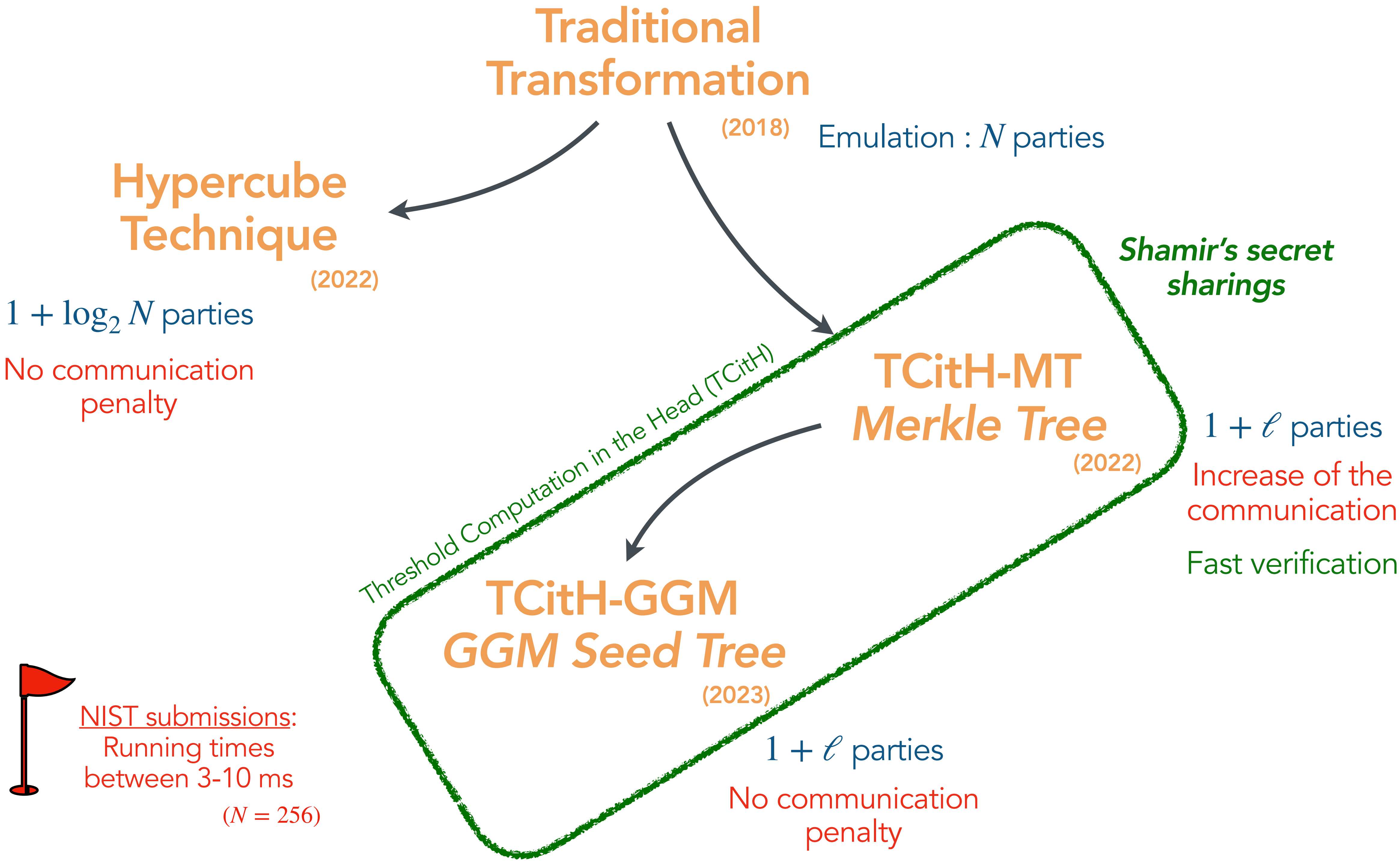
Shamir's secret sharing: to share a value s,

- Build a random degree-ℓ polynomial $P(X) := s + \sum_{j=1}^{\ell} r_j X^j$.
- Set the i^{th} share $[\![s]\!]_i$ as $[\![s]\!]_i := P(e_i)$, where $e_i \neq 0$.

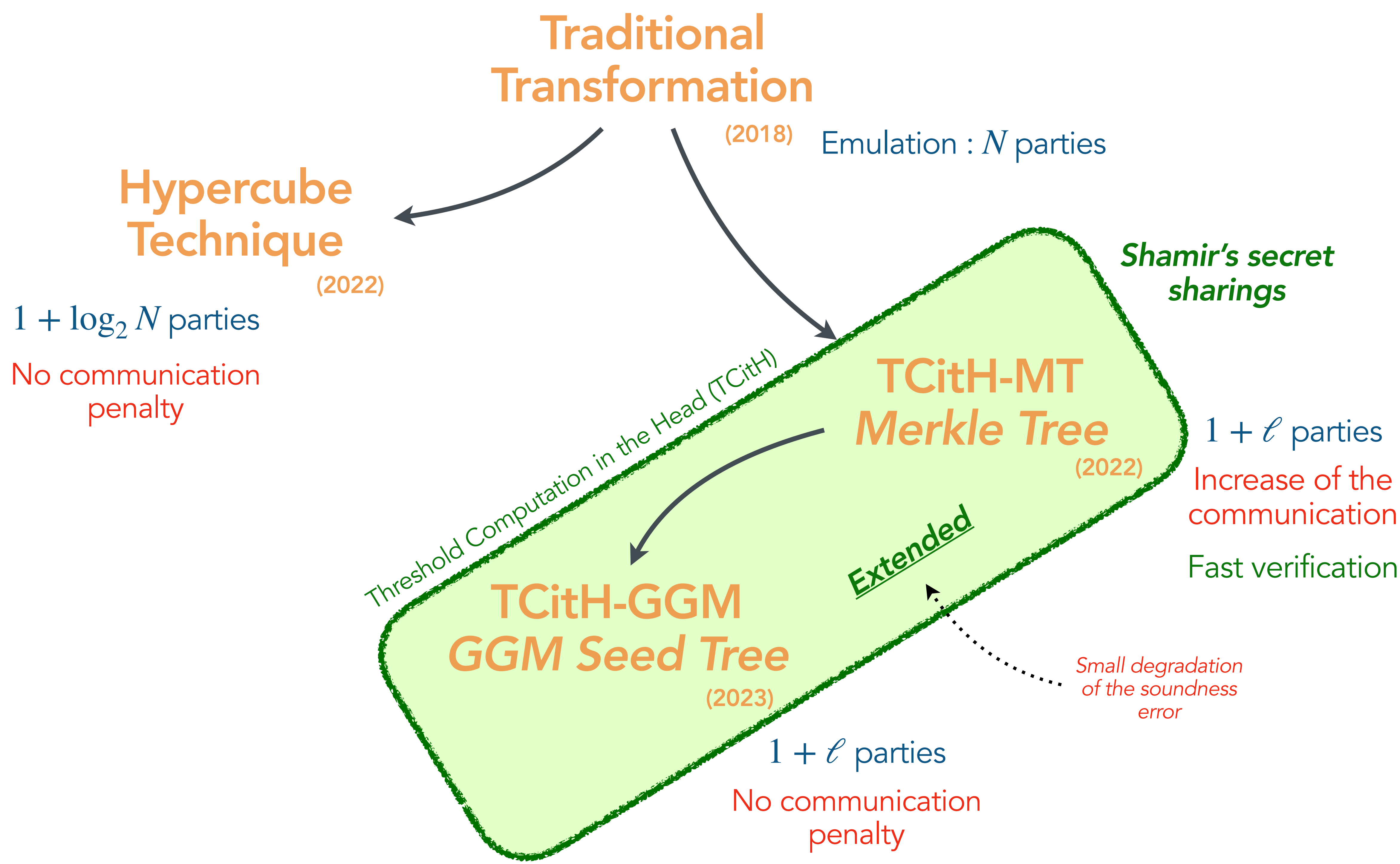
[FR22] Feneuil, Rivain: "Threshold Linear Secret Sharing to the Rescue of MPC-in-the-Head" (ePrint 2022/1407, Asiacrypt 2023)



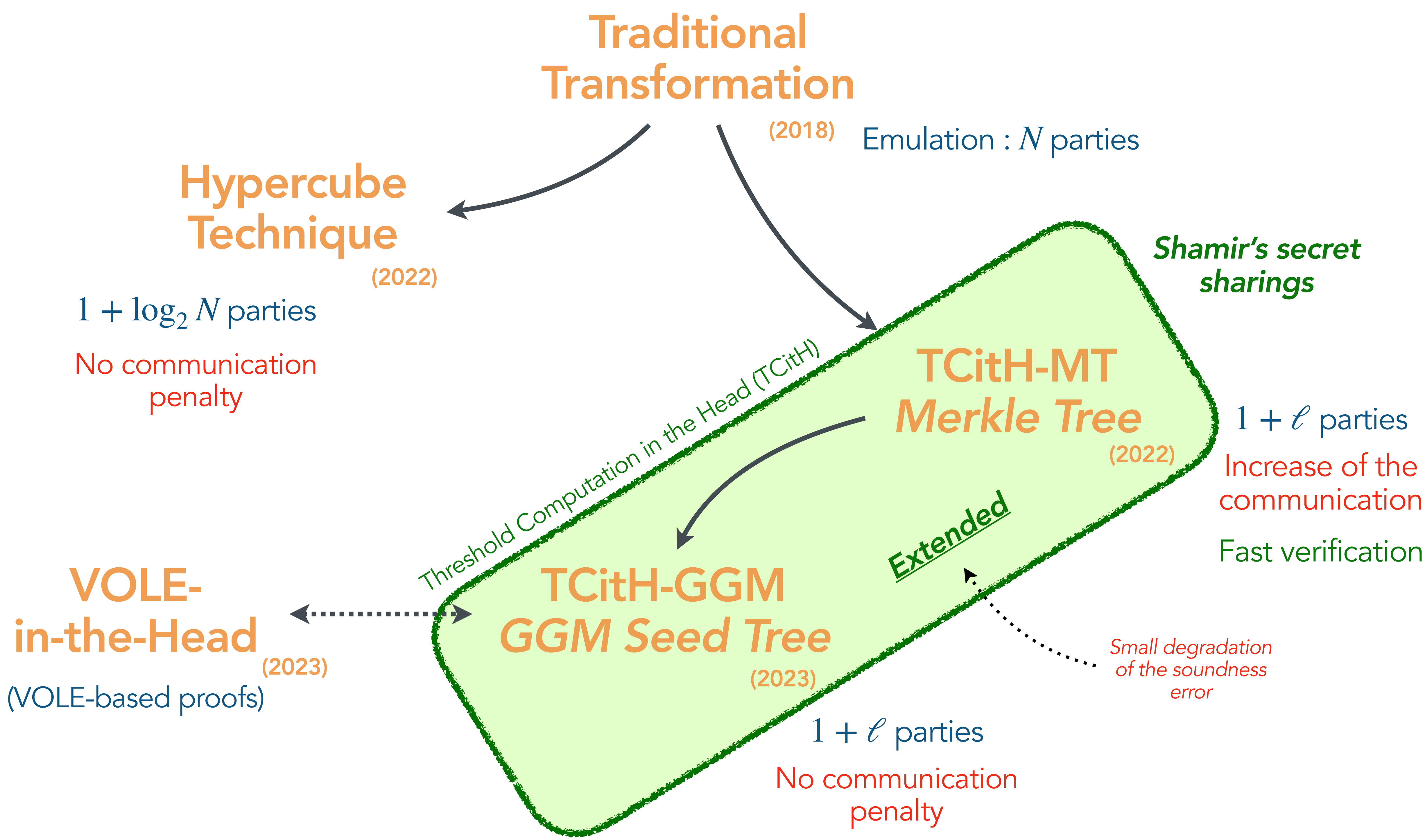
[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)



[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)



[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)



[BBDKORS23] Baum, Braun, Delpech, Klooß, Orsini, Roy, Scholl: "Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures and VOLE-in-the-Head" (Crypto 2023)

Extended TCitH: some applications

[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)

Extended TCitH: some applications

- More efficient signature schemes
  - *Unstructured multivariate quadratic (MQ) problem over* $\mathbb{F}_{251}$
    - MQOM: **6.5 KB**
    - Extended TCitH: **4.2 KB**

[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)

Extended TCitH: some applications

- More efficient signature schemes
  - *Unstructured multivariate quadratic (MQ) problem over* $\mathbb{F}_{251}$
    - MQOM: **6.5 KB**
    - Extended TCitH: **4.2 KB**

- Shorter post-quantum ring signature schemes
  - Extended TCitH with MQ: **5.8 KB** in around 8 ms, for 4000 users
  - Extended TCitH with SD: **10.30 KB** in around 10 ms, for 4000 users

[FR23] Feneuil, Rivain: "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments" (ePrint 2023/1573)

Conclusion

- MPC-in-the-Head
  - Very versatile and tunable
  - A practical tool to build *conservative* signature schemes
    - *Between 4-10 KB in few milliseconds (NIST Level I)*

- Perspectives
  - Active research field
  - Signatures with advanced functionalities:

ring signatures, threshold signatures, multi-signatures,

blind signatures, …

PhD Defense

- Title:

  Post-Quantum Signatures
  from Secure Multiparty Computation

- When: Monday 23rd October 2023, at 2 pm
- Where: Sorbonne University (Jussieu)

More information at

https://www.thibauld-feneuil.fr/phd-defense.html