The AIMer Signature Scheme

Seongkwang Kim[1]
Jihoon Cho[1]
Mingyu Cho[1]
Jincheol Ha[2]
Jihoon Kwon[1]
Byeonghak Lee[1]
Joohee Lee[3]
Jooyoung Lee[2]
Sangyub Lee[1]
Dukjae Moon[1]
Mincheol Son[2]
Hyojin Yoon[1]

[1] Samsung SDS, Seoul, Korea
[2] KAIST, Daejeon, Korea
[3] Sungshin Women's University, Seoul, Korea

Oxford PQC Summit 2023

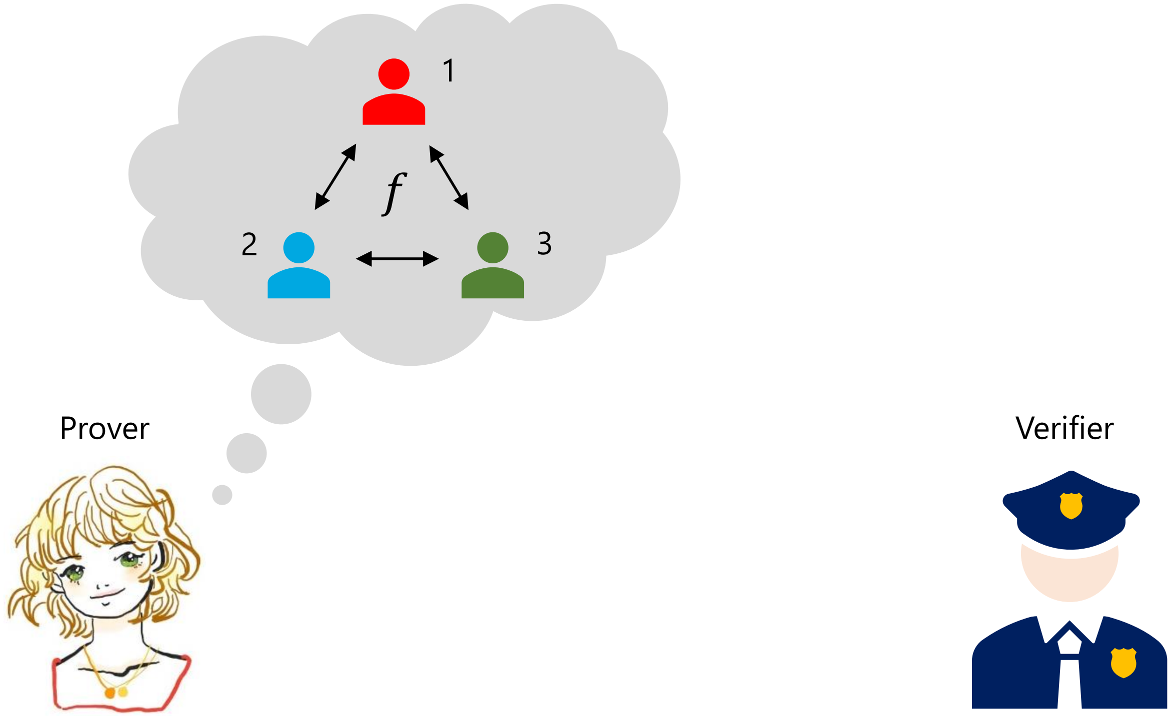
MPC-in-the-Head Paradigm

MPC-in-the-Head Paradigm

- Ishai et al. proposed a generic conversion from MPC to ZKP
- Prover simulates a multiparty computation in her head

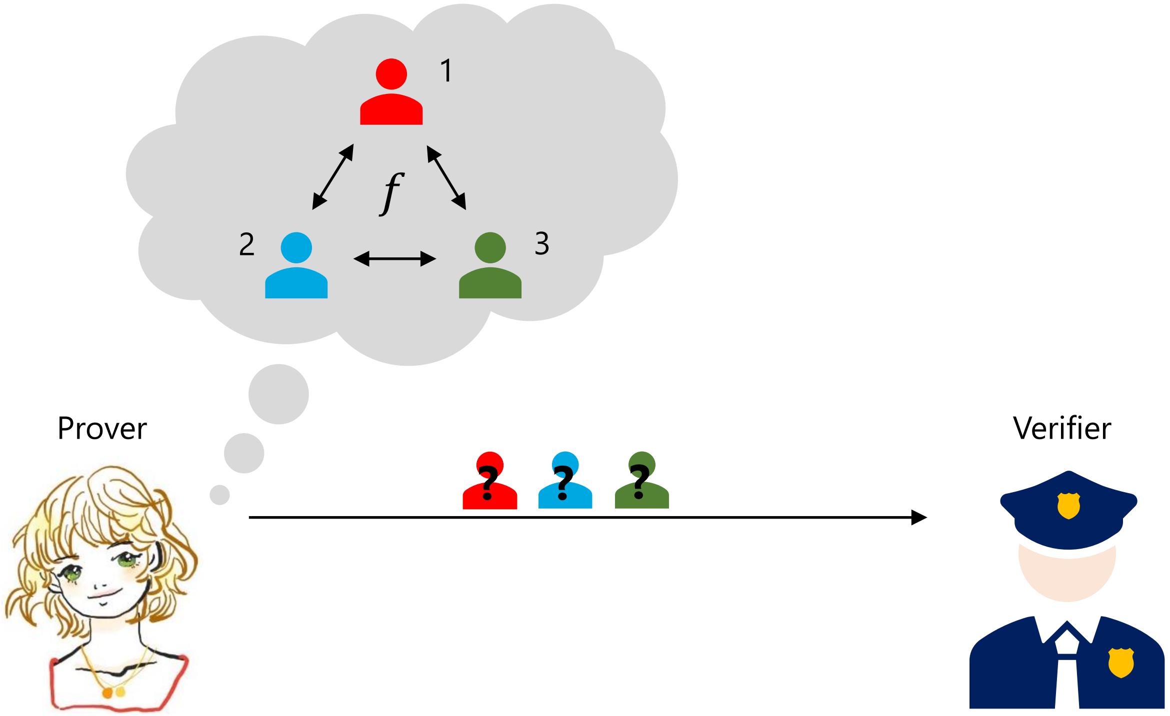
MPC-in-the-Head Paradigm

- Ishai et al. proposed a generic conversion from MPC to ZKP
- Prover simulates a multiparty computation in her head
    1. Prover simulates a multiparty computation of a function f



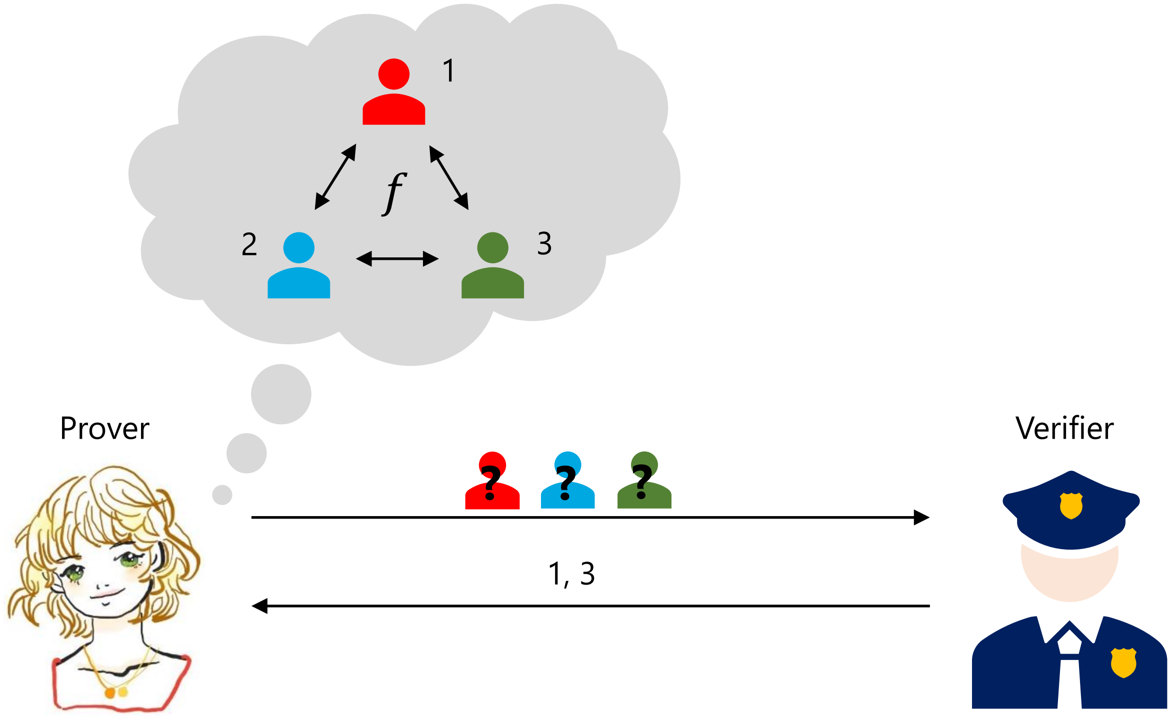
MPC-in-the-Head Paradigm

- Ishai et al. proposed a generic conversion from MPC to ZKP
- Prover simulates a multiparty computation in her head
  1. Prover simulates a multiparty computation of a function f
  2. Prover commits to all the views of the parties



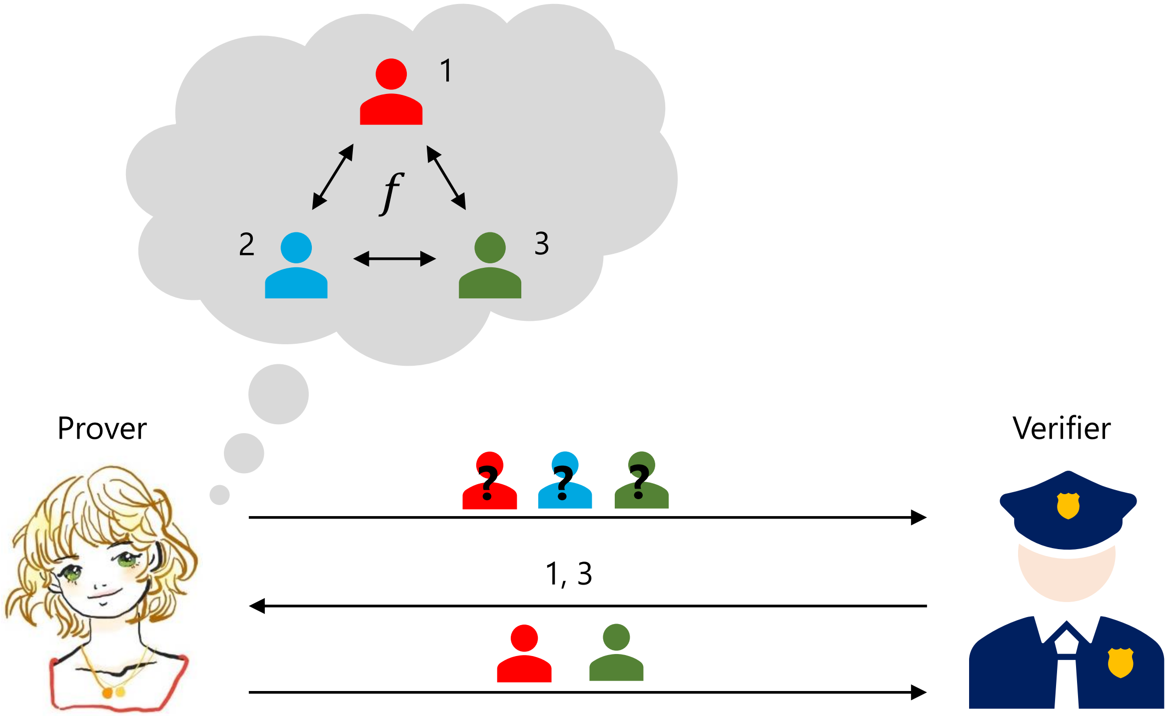
MPC-in-the-Head Paradigm

- Ishai et al. proposed a generic conversion from MPC to ZKP
- Prover simulates a multiparty computation in her head
    1. Prover simulates a multiparty computation of a function f
    2. Prover commits to all the views of the parties
    3. Verifier sends a random challenge



MPC-in-the-Head Paradigm

- Ishai et al. proposed a generic conversion from MPC to ZKP
- Prover simulates a multiparty computation in her head
    1. Prover simulates a multiparty computation of a function f
    2. Prover commits to all the views of the parties
    3. Verifier sends a random challenge
    4. Prover opens the challenged view
    5. Verifier checks consistency



BN++ Proof System

- Kales and Zaverucha proposed an MPCitH-based proof system BN++
- Requires only random oracle and one-way function

BN++ Proof System

- Kales and Zaverucha proposed an MPCitH-based proof system BN++
- Requires only random oracle and one-way function
- Sacrificing-based inner product check
  - Wants to check multiplication triples $\{(x_i, y_i, z_i)\}_i$ such that $x_i \cdot y_i = z_i$
  - Inner product triple $\left((a_i, y_i), c\right)$ such that $\sum_i a_i y_i = c$
  - For random $\{\varepsilon_i\}_i$,

$$[\alpha_i] = \varepsilon_i \cdot [x_i] + [a_i]$$

Open α_i

$$[v] = \sum_i (\alpha_i [y_i] - \varepsilon_i [z_i]) + [c]$$

Check $v = 0$

  - Soundness = $1/|\mathbb{F}|$

Efficient Circuit for BN++

- Arithmetic in a large field (of size $\approx \lambda$)
    - Small field has a poor soundness
- Small number of multiplications (linear maps are free)

Efficient Circuit for BN++

- Arithmetic in a large field (of size $\approx \lambda$)
  - Small field has a poor soundness
- Small number of multiplications (linear maps are free)
- Repeated multiplier
  - If the same multiplier is repeated, the signer can save the signature size
  - $x_1 \cdot y = z_1, x_2 \cdot y = z_2$

Efficient Circuit for BN++

- Arithmetic in a large field (of size $\approx \lambda$)
  - Small field has a poor soundness
- Small number of multiplications (linear maps are free)
- Repeated multiplier
  - If the same multiplier is repeated, the signer can save the signature size
  - $x_1 \cdot y = z_1$, $x_2 \cdot y = z_2$
- Known output share
  - If an output of a multiplication is already known, then the signer can save the signature size
  - E.g., $y = x^{-1} \rightarrow xy = 1$, 1 is known without any computation

Symmetric Primitive AIM

Motivation

- MPC(itH)-friendly symmetric primitives are advanced in directions of:
  - S-boxes on large field
  - Low multiplicative complexity

Motivation

- MPC(itH)-friendly symmetric primitives are advanced in directions of:
  - S-boxes on large field
  - Low multiplicative complexity

- Some symmetric primitives based on large S-boxes have been broken by algebraic attacks
  - MiMC (AC 16, AC 20)
  - Agrasta (C 18, AC 21)
  - Jarvis/Friday (ePrint 18, AC 19)
  - Chaghri (CCS 22, EC 23)

Motivation

- MPC(itH)-friendly symmetric primitives are advanced in directions of:
  - S-boxes on large field
  - Low multiplicative complexity

- Some symmetric primitives based on large S-boxes have been broken by algebraic attacks
  - MiMC (AC 16, AC 20)
  - Agrasta (C 18, AC 21)
  - Jarvis/Friday (ePrint 18, AC 19)
  - Chaghri (CCS 22, EC 23)

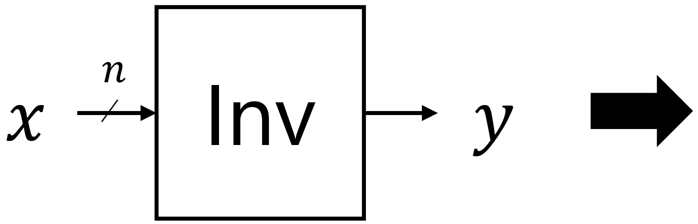
Sufficient security against algebraic attacks + Best performance when combined to BN++

Inverse S-box

- Inverse S-box ($x \mapsto x^{-1}$) is widely used in MPC/ZKP-friendly ciphers
  - High degree, but quadratic relation ($xy = 1$)
  - Invertible
  - Nice DC/LC resistance
  - But, produces many linearly independent quadratic equations

Inverse S-box

- Inverse S-box ($x \mapsto x^{-1}$) is widely used in MPC/ZKP-friendly ciphers
  - High degree, but quadratic relation ($xy = 1$)
  - Invertible
  - Nice DC/LC resistance
  - But, produces many linearly independent quadratic equations



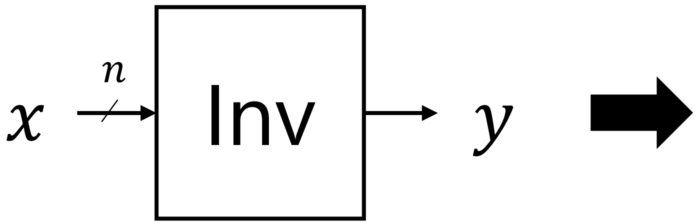
$$\begin{cases} f_1(x_1, \dots, x_n, y_1, \dots, y_n) = 0 \\ \vdots \\ f_{5n}(x_1, \dots, x_n, y_1, \dots, y_n) = 0 \end{cases}$$

$5n$ quadratic equations

c.f. optimally n equations

Inverse S-box

- Inverse S-box ($x \mapsto x^{-1}$) is widely used in MPC/ZKP-friendly ciphers
  - High degree, but quadratic relation ($xy = 1$)
  - Invertible
  - Nice DC/LC resistance
  - But, produces many linearly independent quadratic equations



$$\begin{cases} f_1(x_1, \dots, x_n, y_1, \dots, y_n) = 0 \\ \vdots \\ f_{5n}(x_1, \dots, x_n, y_1, \dots, y_n) = 0 \end{cases}$$

$5n$ quadratic equations

c.f. optimally n equations

More equations lead to a weaker resistance against algebraic attacks!

Candidates of Appropriate S-box

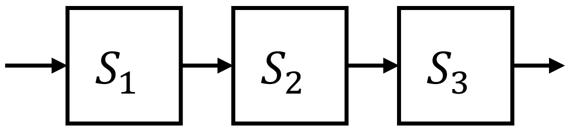
- Niho exponent
  - $x \mapsto x^{2^s+2^{s/2}-1}$ over $\mathbb{F}_{2^n}$, $n = 2s + 1$
  - n equations, high-degree
  - 2 multiplications, odd-length field
- NGG exponent (Nawaz et al., 2009)
  - $x \mapsto x^{2^{s+1}+2^{s-1}-1}$ over $\mathbb{F}_{2^n}$, $n = 2s$
  - $2n$ equations, even-length field, good DC/LC resistance
  - 2 multiplications
- Mersenne exponent
  - $x \mapsto x^{2^s-1}$ over $\mathbb{F}_{2^n}$
  - $3n$ equations, even-length field, single multiplication
  - moderate DC/LC resistance
- Gold exponent
  - $x \mapsto x^{2^s+1}$ over $\mathbb{F}_{2^n}$
  - Even-length field, single multiplication, good DC/LC resistance
  - $4n$ equations

Repetitive Structure for BN++

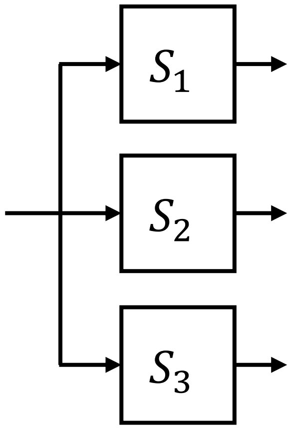
- Repeated multiplier technique (in BN++)
  - If prover needs to check multiple multiplications with a same multiplier,
    - e.g., $x_1 \cdot y = z_1, x_2 \cdot y = z_2$
  - Then, the prover can prove them in a batched way
  - More same multiplier → Smaller signature size

Repetitive Structure for BN++

- Repeated multiplier technique (in BN++)
  - If prover needs to check multiple multiplications with a same multiplier,
    - e.g., $x_1 \cdot y = z_1, x_2 \cdot y = z_2$
  - Then, the prover can prove them in a batched way
  - More same multiplier → Smaller signature size

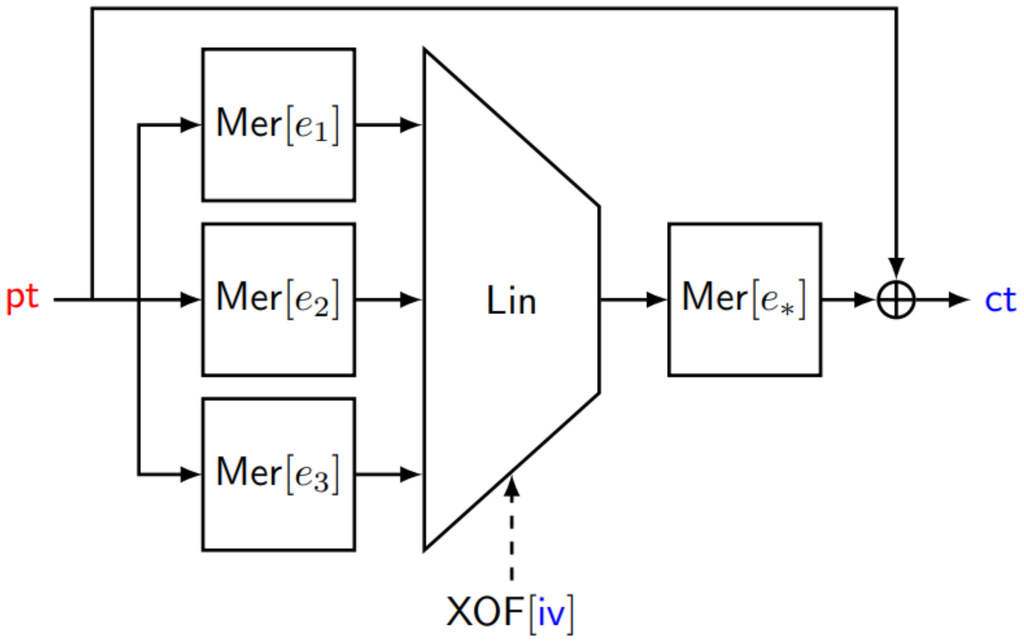


Serial S-box
(Limited application of repeated multiplier)



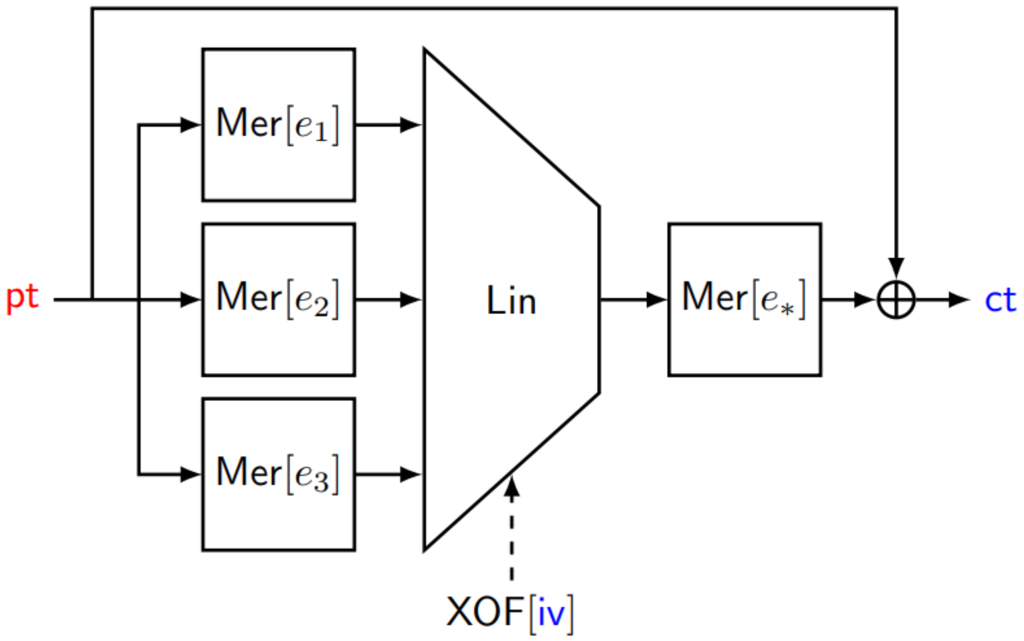
Parallel S-box
(Full application of repeated multiplier)

Symmetric Primitive AIM



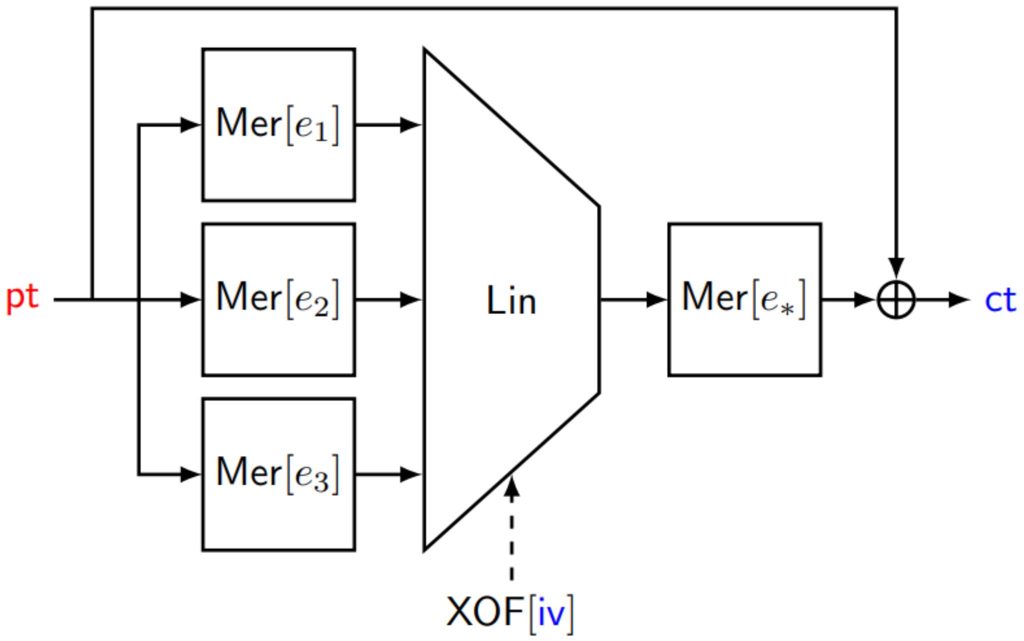
- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance

Symmetric Primitive AIM



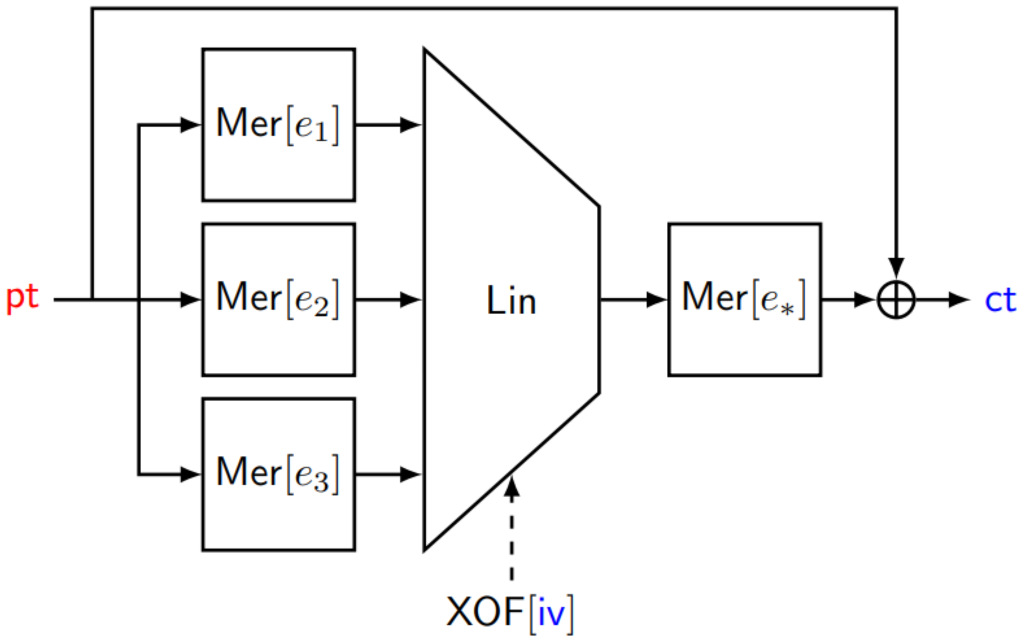
- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance
- Repetitive structure
  - Parallel application of S-boxes
  - Feed-forward construction
  - Fully exploit the BN++ optimizations
  - Locally-computable output share

Symmetric Primitive AIM



- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance
- Repetitive structure
  - Parallel application of S-boxes
  - Feed-forward construction
  - Fully exploit the BN++ optimizations
  - Locally-computable output share
- Randomized structure
  - Affine layer is generated from XOF

Symmetric Primitive AIM



Scheme	λ	n	ℓ	e_1	e_2	e_3	e_*
AIM-I	128	128	2	3	27	-	5
AIM-III	192	192	2	5	29	-	7
AIM-V	256	256	3	3	53	7	5

- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance
- Repetitive structure
  - Parallel application of S-boxes
  - Feed-forward construction
  - Fully exploit the BN++ optimizations
  - Locally-computable output share
- Randomized structure
  - Affine layer is generated from XOF

Analyses on AIM

Recent Analysis on AIM (Jul. 24)

- Fukang Liu and Mohammad Mahzoun proposed a fast exhaustive search attack on AIM*
- It achieves 10-12 bits smaller complexity compared to brute-force attack on AIM
- The main vulnerability was that there are low-degree equations with n Boolean variables
- Increasing exponents resolves this vulnerability

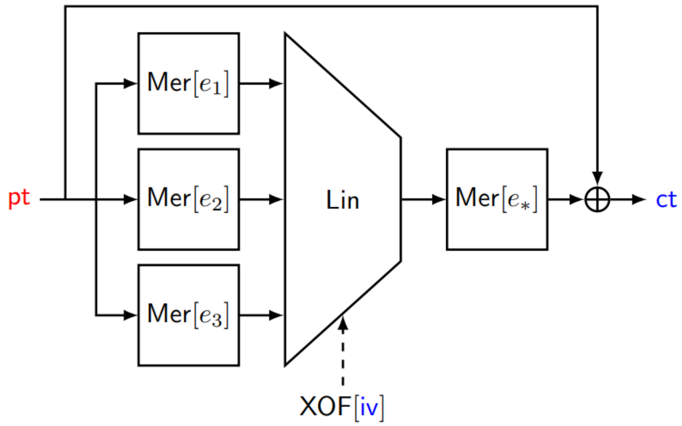
* F. Liu and M. Mahzoun. "Algebraic Attacks on RAIN and AIM Using Equivalent Representations". Cryptology ePrint Archive. Report 2023/1133

Recent Analysis on AIM (Jul. 27)

- Liu introduce another possible vulnerability to our team*
- Setting a new variable $w = \mathrm{pt}^{-1}$ leads to easier system than expected
  - AIM is claimed to be secure under an ℓn-variable system with $3\ell n$ quadratic equations
  - A $2n$-variable system including $5n$ quadratic equations and $5n$ cubic equations

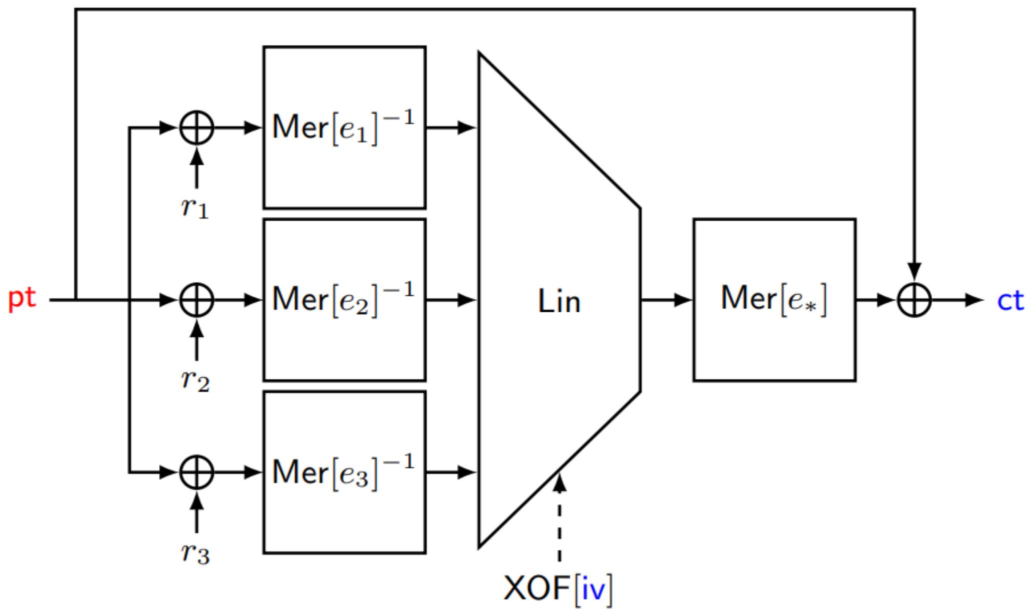
$$\begin{cases} \mathrm{pt} \cdot w = 1 \\ \mathrm{Lin}\left(\mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w\right) \cdot (\mathrm{pt} + \mathrm{ct}) = \mathrm{Lin}\left(\mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w\right)^{2^{e_*}} \\ \mathrm{Lin}\left(\mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w\right) \cdot (1 + w \cdot \mathrm{ct}) = w \cdot \mathrm{Lin}\left(\mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w, \mathrm{pt}^{2^{e_1}}w\right)^{2^{e_*}} \end{cases}$$

- Note that this attack is **not practically feasible** on AIM



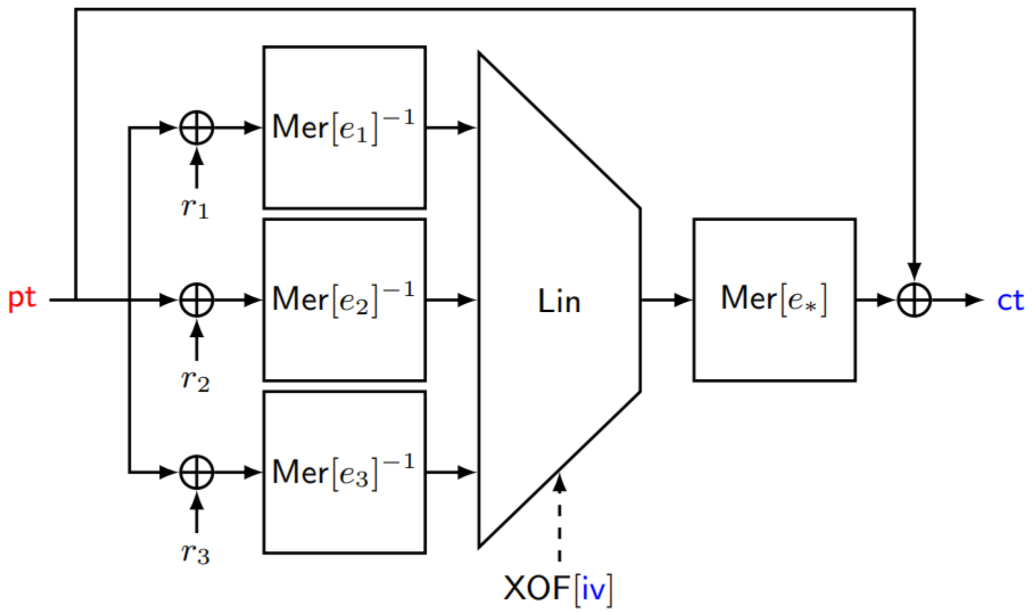
* In private communication

AIM2: Secure Patch for Algebraic Attacks (In Progress)



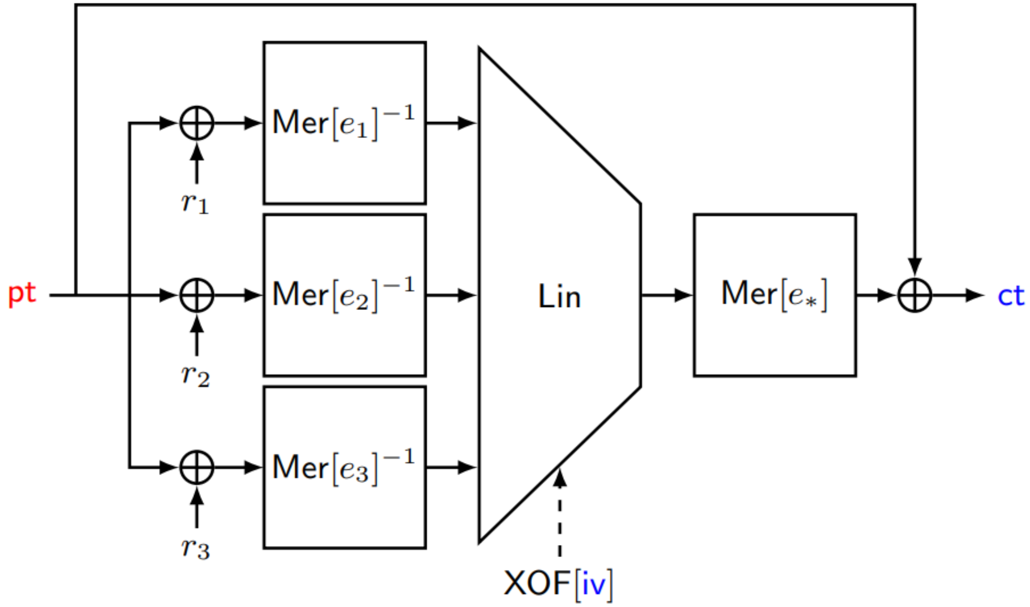
- Inverse Mersenne S-box
  - $\text{Mer}[e]^{-1}(x) = x^a$
  - $a = (2^e - 1)^{-1} \bmod (2^n - 1)$
  - More resistant to algebraic attacks

AIM2: Secure Patch for Algebraic Attacks (In Progress)



- Inverse Mersenne S-box
  - $\mathrm{Mer}[e]^{-1}(x) = x^a$
  - $a = (2^e - 1)^{-1} \bmod (2^n - 1)$
  - More resistant to algebraic attacks
- Larger exponents
  - To mitigate fast exhaustive search

AIM2: Secure Patch for Algebraic Attacks (In Progress)



- Inverse Mersenne S-box
  - $\mathrm{Mer}[e]^{-1}(x) = x^a$
  - $a = (2^e - 1)^{-1} \bmod (2^n - 1)$
  - More resistant to algebraic attacks
- Larger exponents
  - To mitigate fast exhaustive search
- Fixed constant addition
  - To differentiate inputs of S-boxes
  - Increase the degree of composite power function

$$(x^a)^b \text{ vs } (x^a + c)^b$$

Analysis on AIM2

- Algebraic attacks
  - Fast exhaustive search: mitigated by high exponents
  - Brute-force search of quadratic equations
  - Toy experiment of good intermediate variables

Analysis on AIM2

- Algebraic attacks
  - Fast exhaustive search: mitigated by high exponents
  - Brute-force search of quadratic equations
  - Toy experiment of good intermediate variables
- Other attacks
  - Exhaustive key search: slightly increased complexity
  - Very recent analysis (Sep. 2) of Markku is also mitigated (and it will be updated)
  - LC/DC: almost same
  - Quantum attacks: complexities change not critically

Analysis on AIM2

- Algebraic attacks
  - Fast exhaustive search: mitigated by high exponents
  - Brute-force search of quadratic equations
  - Toy experiment of good intermediate variables
- Other attacks
  - Exhaustive key search: slightly increased complexity
  - Very recent analysis (Sep. 2) of Markku is also mitigated (and it will be updated)
  - LC/DC: almost same
  - Quantum attacks: complexities change not critically
- Performance
  - Signature size: exactly the same
  - Sign/verify time: about 10% increase

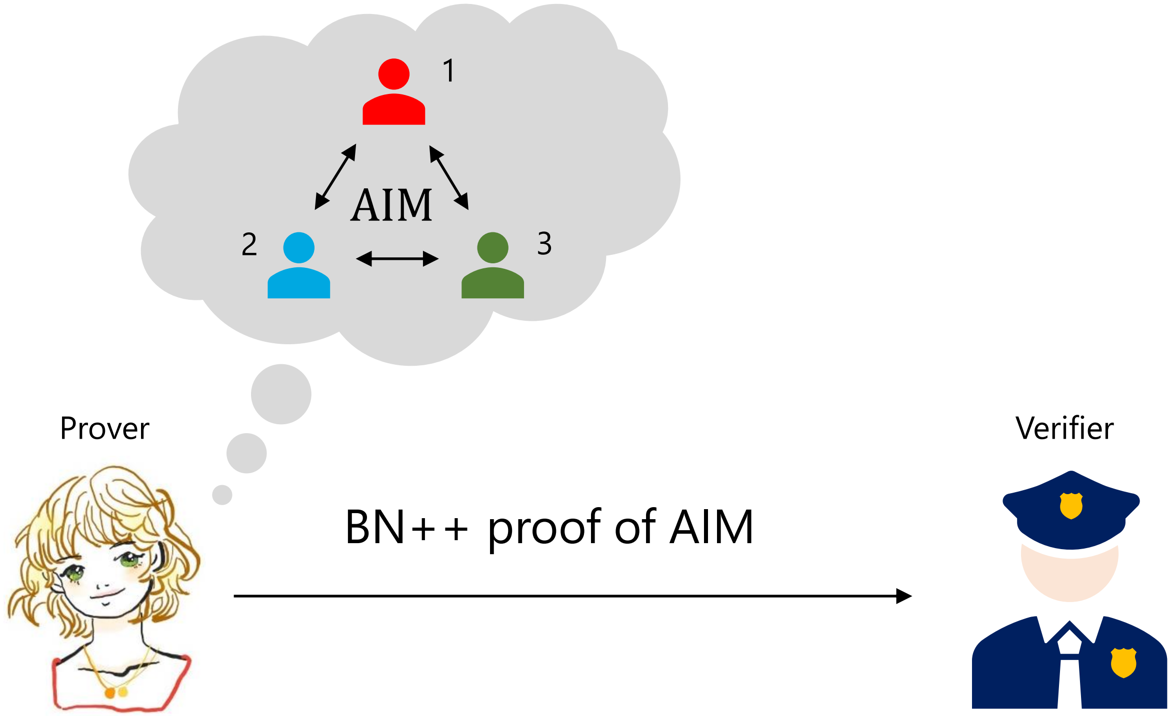
Analysis on AIM2

- Algebraic attacks
  - Fast exhaustive search: mitigated by high exponents
  - Brute-force search of quadratic equations
  - Toy experiment of good intermediate variables
- Other attacks
  - Exhaustive key search: slightly increased complexity
  - Very recent analysis (Sep. 2) of Markku is also mitigated (and it will be updated)
  - LC/DC: almost same
  - Quantum attacks: complexities change not critically
- Performance
  - Signature size: exactly the same
  - Sign/verify time: about 10% increase
- Preliminary version can be found in our website!

The AIMer Signature Scheme

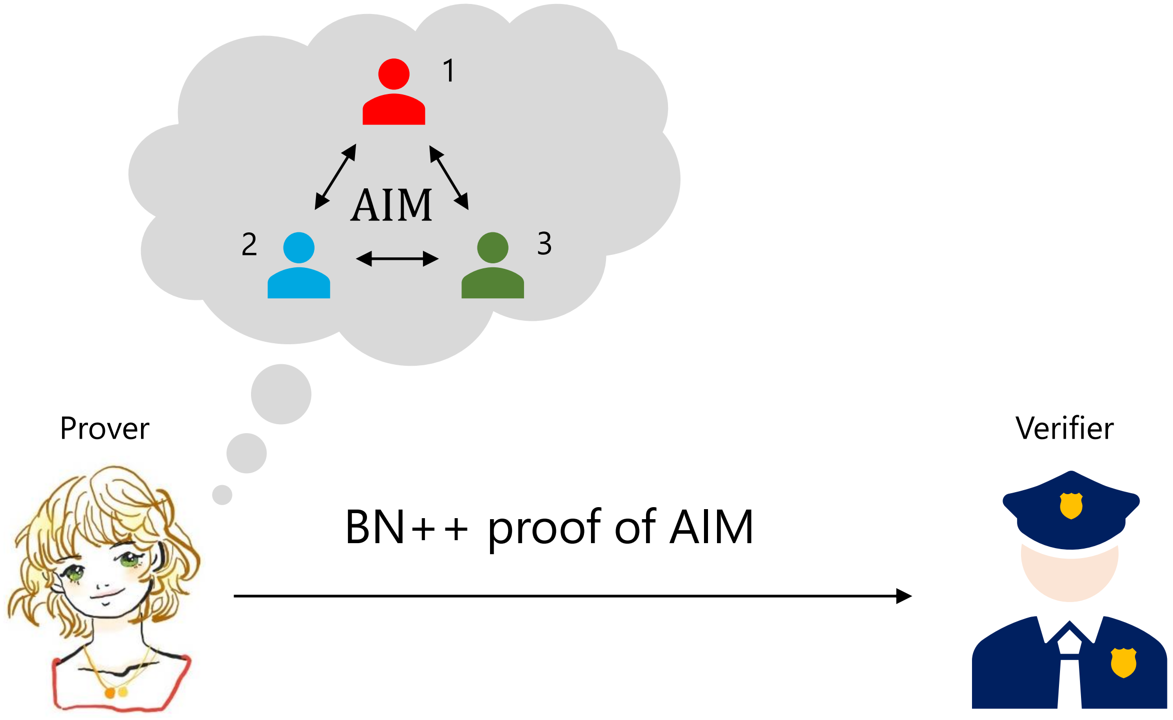
AIMer Signature Scheme

- AIMer = BN++ proof of knowledge of AIM input
- Security is based on the one-wayness of AIM in the ROM



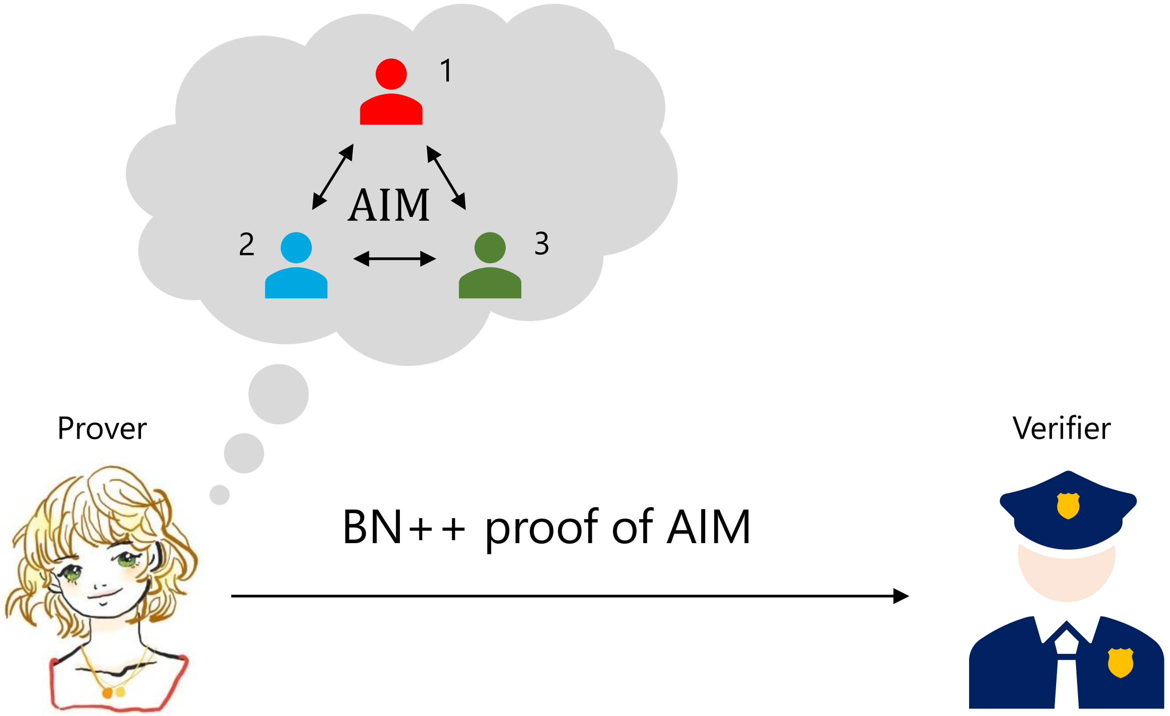
AIMer Signature Scheme

- AIMer = BN++ proof of knowledge of AIM input
- Security is based on the one-wayness of AIM in the ROM
- Advantages
    - Security based on only symmetric primitives
    - Fast key generation
    - Small key sizes
    - Trade-offs between signatures size and speed
    - Randomness misuse resistance



AIMer Signature Scheme

- AIMer = BN++ proof of knowledge of AIM input
- Security is based on the one-wayness of AIM in the ROM
- Advantages
  - Security based on only symmetric primitives
  - Fast key generation
  - Small key sizes
  - Trade-offs between signatures size and speed
  - Randomness misuse resistance
- Limitations
  - Newly-designed symmetric primitive AIM
  - Moderately large signature size (3.8~5.9 KB)
  - Slow signing/verifying speed (0.59~22 ms)



Performance Comparison

Scheme	pk (B)	sig (B)	Sign (ms)	Verify (ms)
Dilithium2	1312	2420	0.10	0.03
Falcon-512	897	690	0.27	0.04
SPHINCS$^+$-128s	32	7856	315.74	0.35
SPHINCS$^+$-128f	32	17088	16.32	0.97
Picnic1-L1-full	32	30925	1.16	0.91
Picnic3	32	12463	5.83	4.24
Banquet	32	19776	7.09	5.24
Rainier$_3$	32	8544	0.97	0.89
BN++Rain$_3$	32	6432	0.83	0.77
AIMer-L1 (Not updated)	32	5904	0.59	0.53
AIMer-L1 (Not updated)	32	3840	22.29	21.09

Some Remarks

- Remark
  - We submitted AIMer to KpqC and NIST PQC competition
  - Our homepage: https://aimer-signature.org
  - We are waiting for **third-party analysis**!

- Future work
  - Updates on the specification document
  - QROM security of AIMer
  - More optimization on BN++

Thank you!
Check out our website!