AIM: Symmetric Primitive for Shorter Signatures with Stronger Security

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

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

ACM CCS 2023

SAMSUNG SDS KAIST SUNGSHIN WOMEN'S UNIVERSITY

Brief Overview

- Background
  - MPC-in-the-Head (MPCitH) paradigm is a conversion from MPC to ZKP
  - A signature scheme is obtained if MPCitH is combined with Fiat-Shamir transform and OWF

Brief Overview

- Background
  - MPC-in-the-Head (MPCitH) paradigm is a conversion from MPC to ZKP
  - A signature scheme is obtained if MPCitH is combined with Fiat-Shamir transform and OWF

- Contribution
  - We propose symmetric primitive AIM for shorter MPCitH-based signatures
  - We reduce signature size by $\geq$8% compared to previous MPCitH-based signature schemes

Brief Overview

- Background
  - MPC-in-the-Head (MPCitH) paradigm is a conversion from MPC to ZKP
  - A signature scheme is obtained if MPCitH is combined with Fiat-Shamir transform and OWF
- Contribution
  - We propose symmetric primitive AIM for shorter MPCitH-based signatures
  - We reduce signature size by ≥8% compared to previous MPCitH-based signature schemes
- Amendment
  - Recently, there have been multiple analyses on AIM
  - We patched AIM to AIM2 without significant performance degradation

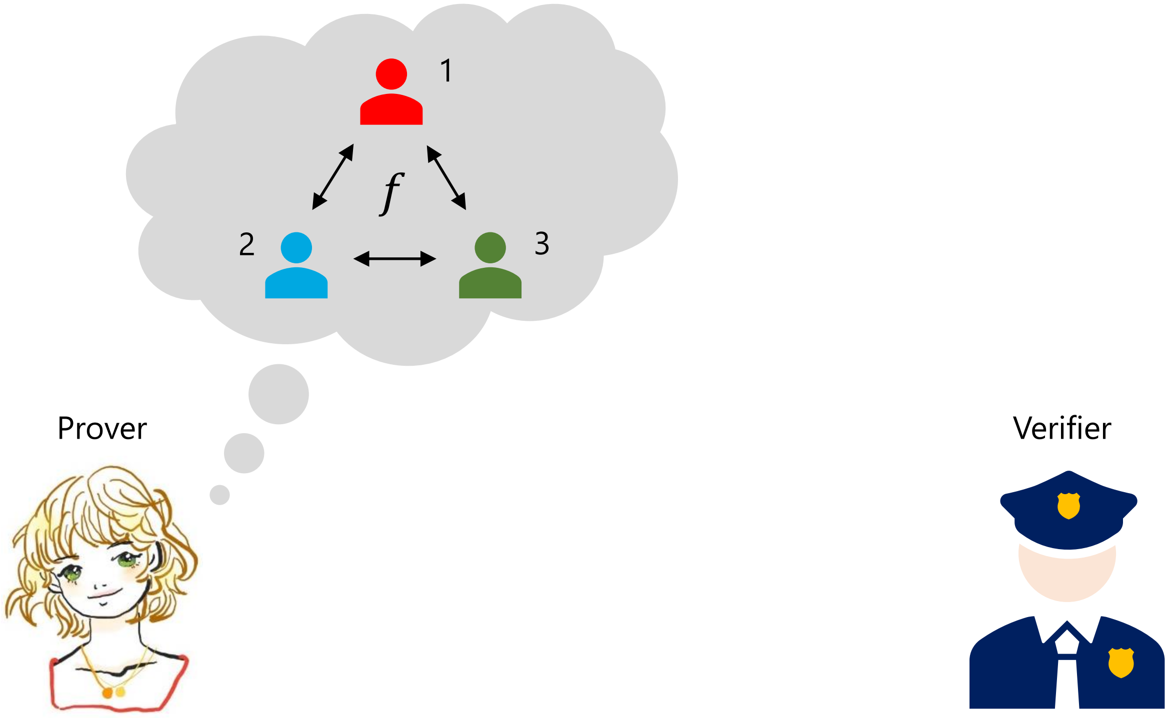
MPC-in-the-Head Paradigm

MPC-in-the-Head Paradigm

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

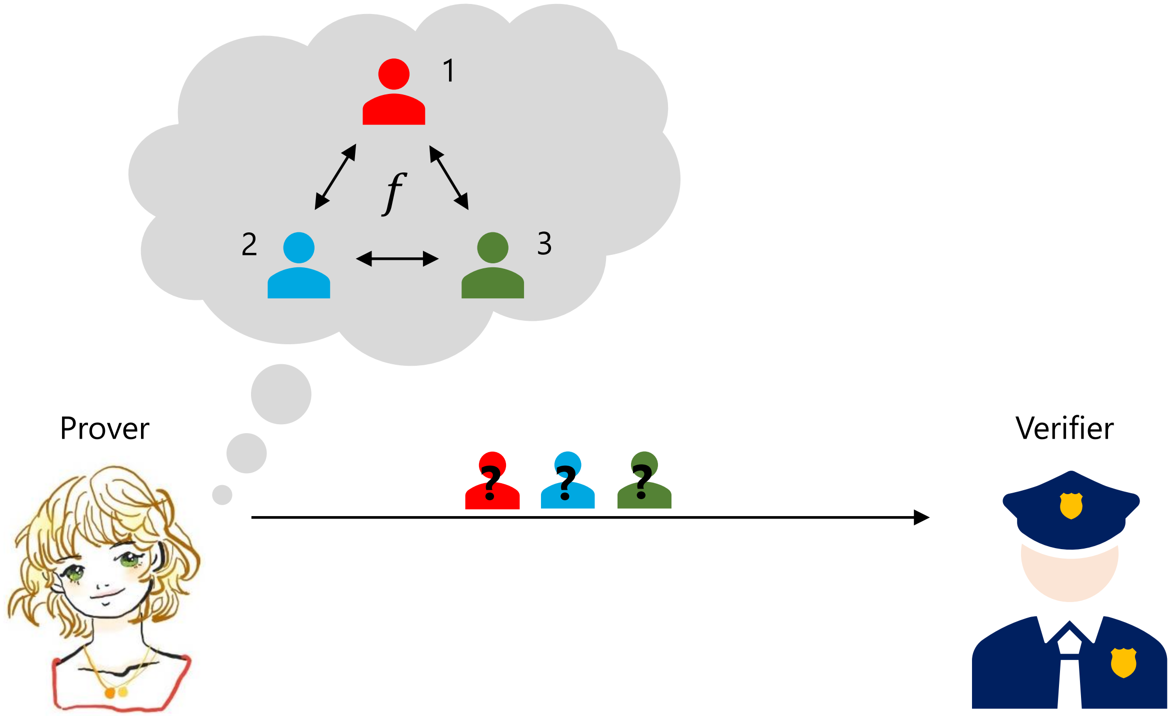
MPC-in-the-Head Paradigm

- 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



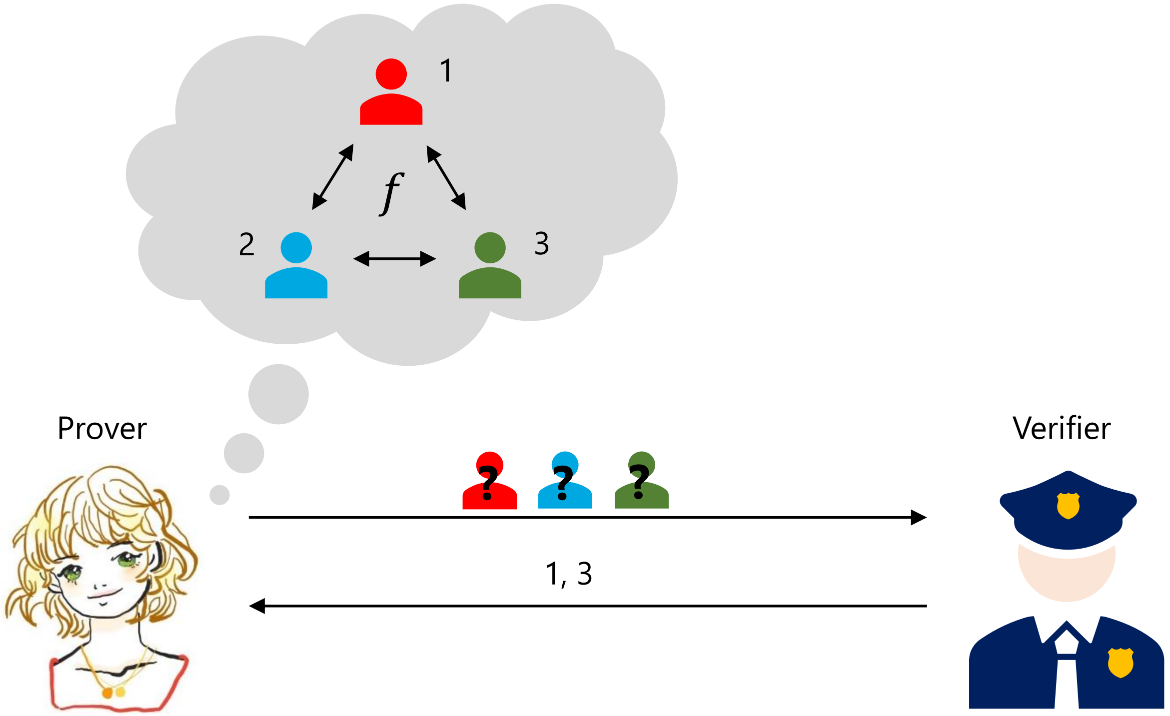
MPC-in-the-Head Paradigm

- 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



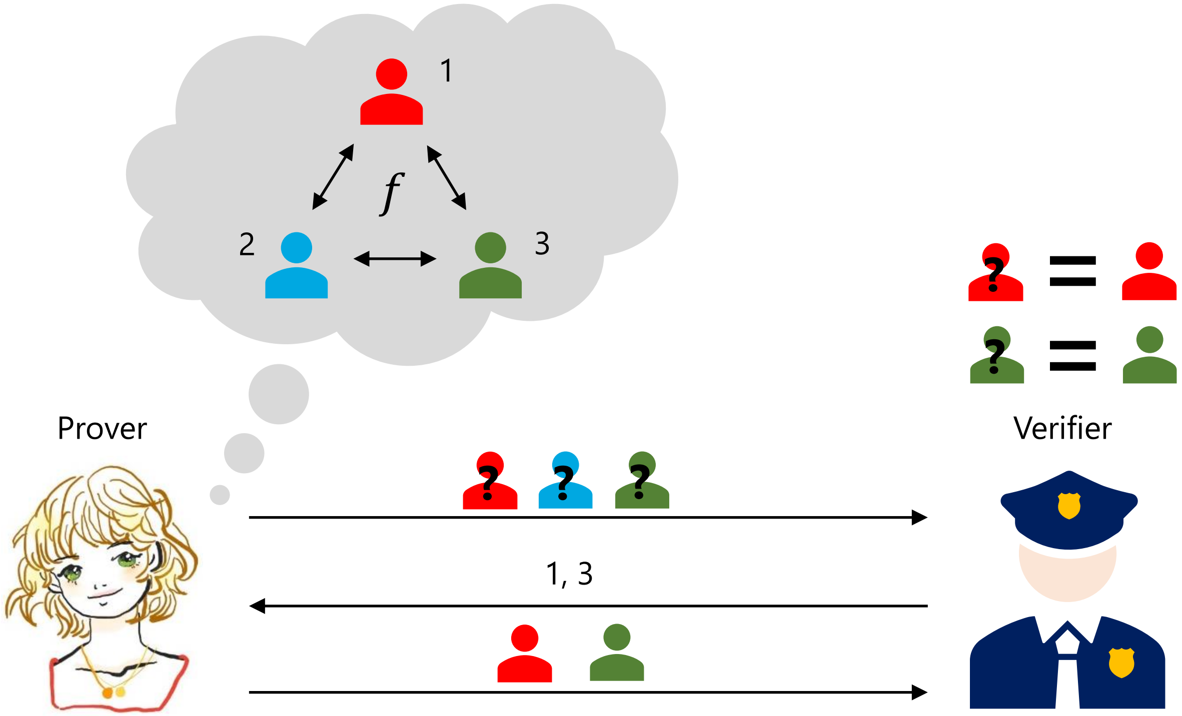
MPC-in-the-Head Paradigm

- 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
- An arithmetic circuit is efficiently provable by BN++ if:
  - Arithmetic is over a large field (of size $\approx \lambda$)
  - Small number of multiplications
  - The same multiplier is repeated ($x_1 \cdot y = z_1$, $x_2 \cdot y = z_2$)
  - An output of a multiplication is already known (e.g., $S(x) = x^{-1} \Rightarrow x \cdot S(x) = 1$)

BN++ Proof System

- Kales and Zaverucha proposed an MPCitH-based proof system BN++
- Requires only random oracle and one-way function
- An arithmetic circuit is efficiently provable by BN++ if:
  - Arithmetic is over a large field (of size $\approx \lambda$)
  - Small number of multiplications
  - The same multiplier is repeated ($x_1 \cdot y = z_1$, $x_2 \cdot y = z_2$)
  - An output of a multiplication is already known (e.g., $y = x^{-1} \Rightarrow xy = 1$)
- Given a one-way function $f(x) = y$, BN++ proof of x becomes a signature scheme

Symmetric Primitive AIM

Motivation

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

Motivation

- MPC(itH)-friendly symmetric primitives are advanced in directions of:
  - S-boxes on a 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 a 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

+

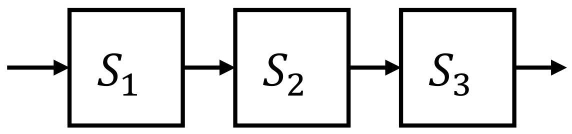
Best performance when combined to BN++

Repetitive Structure for BN++

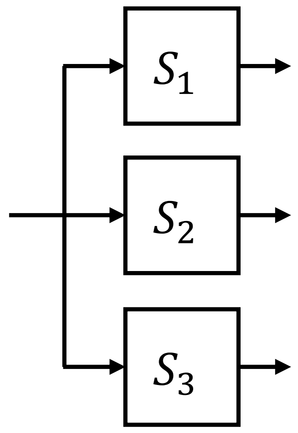
- Repeated multiplier technique (in BN++)
  - If prover needs to check multiple multiplications with a same multiplier,
    - e.g. $x_1 \cdot y = z_1, x_2 \cdot y = z_2$
  - Then, the prover can prove them in a batched way
  - More same multiplier → Smaller signature size

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)

Appropriate Choice of S-box

- Requirements

Security	Efficiency
Invertible	Using large field multiplication
Nice differential/linear properties	Few multiplications to verify
High-degree	(e.g., $S(x) = x^{-1} \Rightarrow x \cdot S(x) = 1$)
Small number of quadratic equations	

Appropriate Choice of S-box

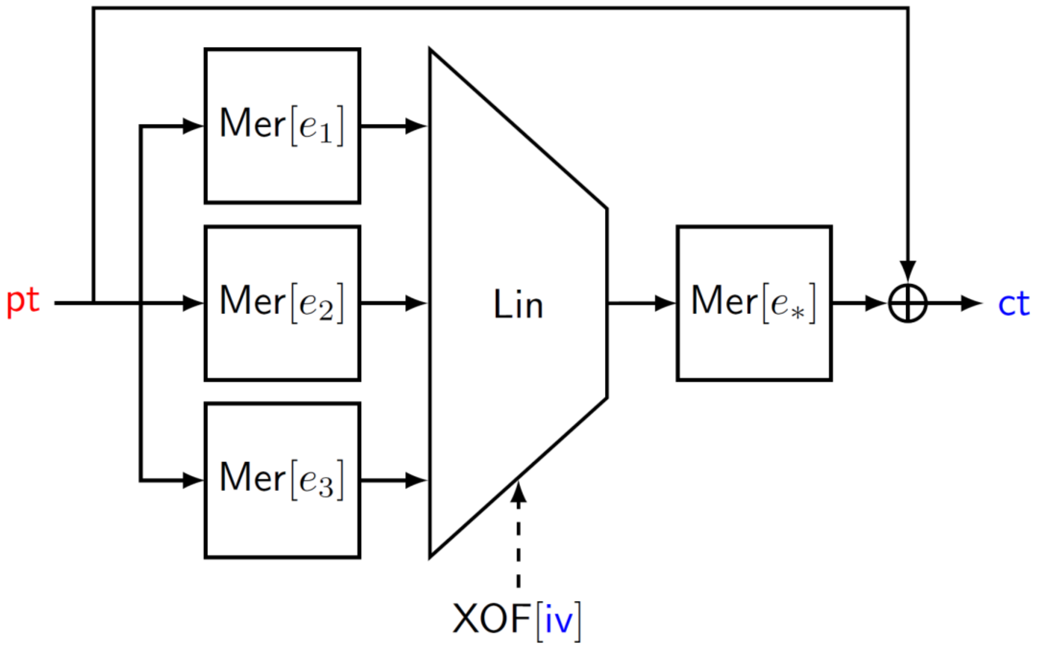
- Requirements

Security	Efficiency
Invertible	Using large field multiplication
Nice differential/linear properties	Few multiplications to verify
High-degree	(e.g., $S(x) = x^{-1} \Rightarrow x \cdot S(x) = 1$)
Small number of quadratic equations	

- Mersenne S-box
  - $\mathrm{Mer}[e](x) = x^{2^e - 1}$

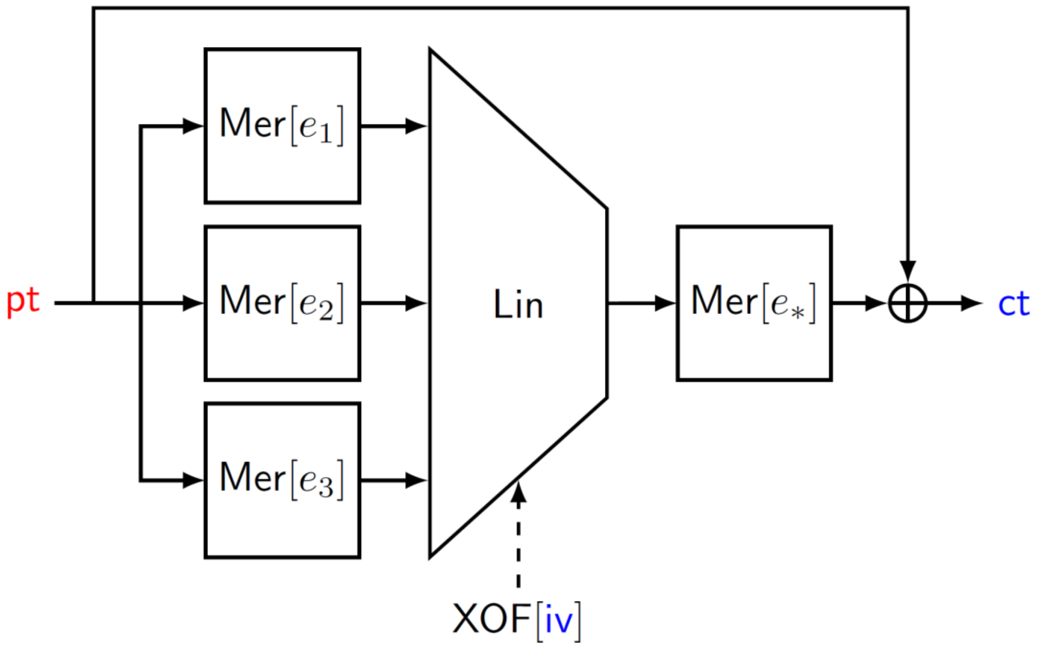
Security	Efficiency
Invertible	$GF(2^\lambda)$ field multiplication
Moderate differential/linear properties	Single multiplication to verify
Degree e	(i.e., $x \cdot S(x) = x^{2^e}$)
$3n$ quadratic equations	

Symmetric Primitive AIM



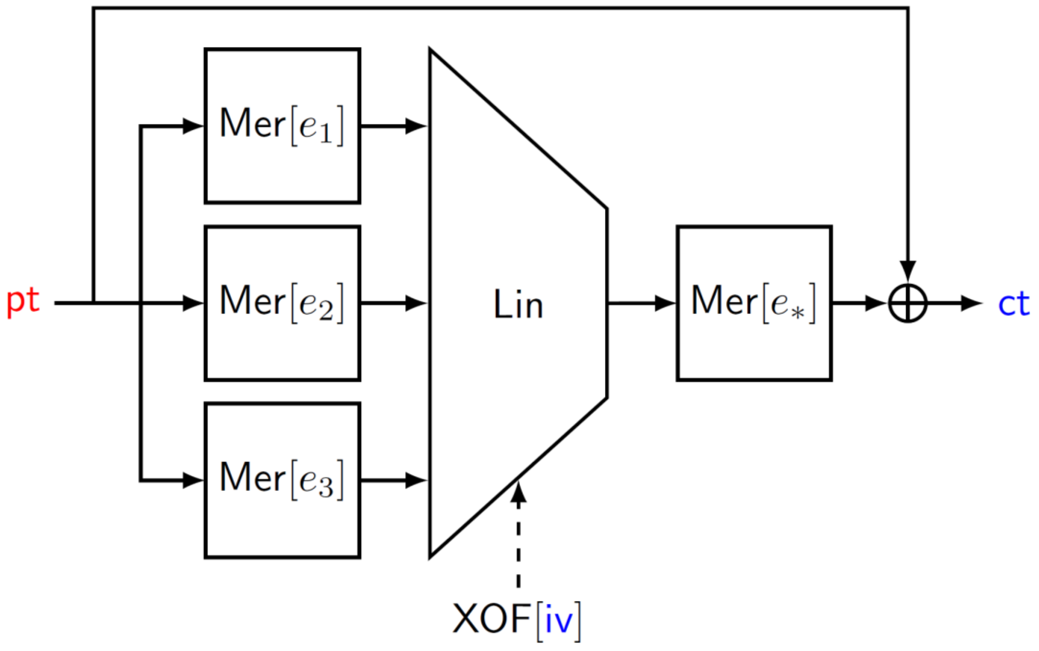
- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance

Symmetric Primitive AIM



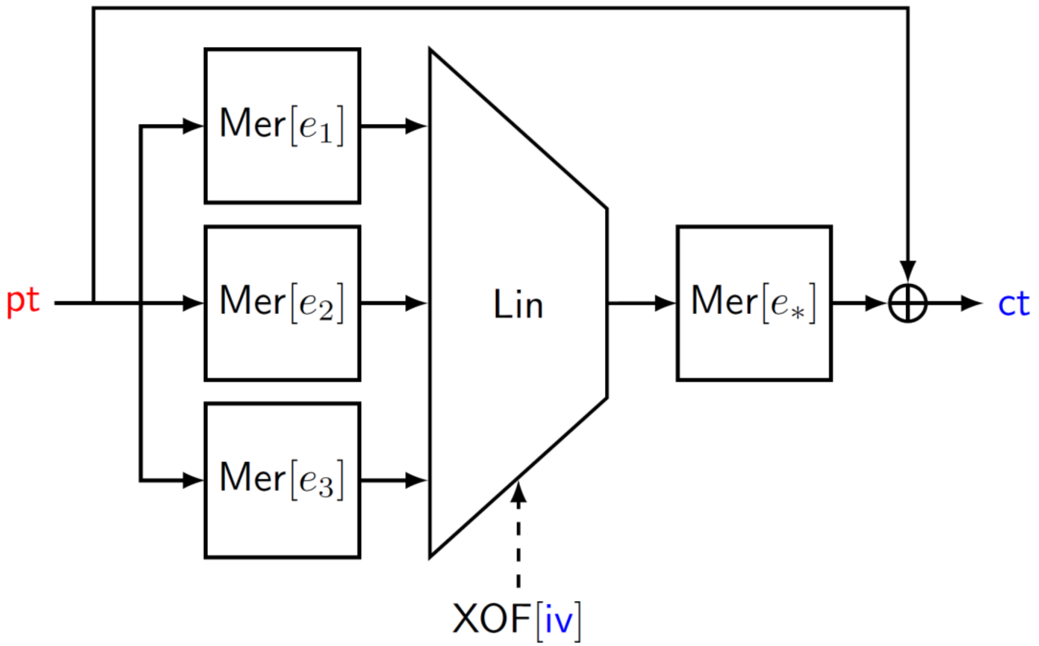
- Mersenne S-box
  - Invertible, high-degree, quadratic relation
  - Requires a single multiplication
  - Produces $3n$ quadratic equations
  - Moderate DC/LC resistance
- Repetitive structure
  - Parallel application of S-boxes
  - Feed-forward construction
  - Fully exploit the BN++ optimizations
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

- Recent analysis on AIM
  - [LMOM23] Fukang Liu et al. Fast exhaustive search, giving up to 12-bit security degradation
  - [Liu23] Less costly algebraic attack, but not broken
  - [Sar23] Efficient key search (by implementation), unknown amount of security degradation
  - [ZWYGC23] Guess & determine + linearization attack, giving up to 6-bit security degradation

[LMOM23] F. Liu, M. Mahzoun, M. Øygarden, and W. Meier. *Algebraic Attacks on RAIN and AIM Using Equivalent Representations*. IACR Cryptology ePrint Archive. Report 2023/1133. https://eprint.iacr.org/2023/1133.

[Liu23] F. Liu. *Mind Multiple Power Maps: Algebraic Cryptanalysis of Full AIM for Post-quantum Signature Scheme AIMer*. In private communication. 2023.

[Sar23] M. O. Saarinen. *Round 1 (Additional Signatures) OFFICIAL COMMENT: AIMer*. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/BI2iIXblNy0.

[ZWYGC23] K. Zhang, Q. Wang, Y. Yu, C. Guo, and H. Cui. *Algebraic Attacks on Round-Reduced RAIN and Full AIM-III*. IACR Cryptology ePrint Archive. Report 2023/1397. https://eprint.iacr.org/2023/1397. To appear Asiacrypt 2023.

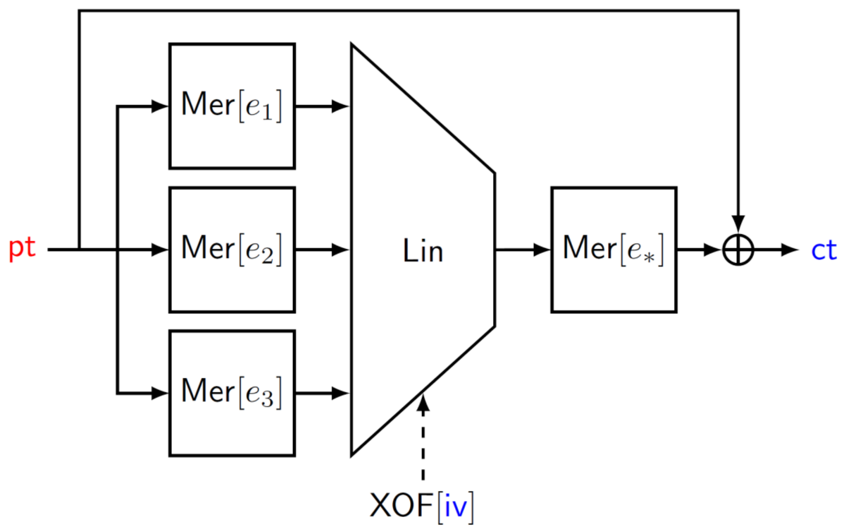
Recent Analysis on AIM

- Recent analysis on AIM
  - [LMOM23] Fukang Liu et al. Fast exhaustive search, giving up to 12-bit security degradation
  - [Liu23] Less costly algebraic attack, but not broken
  - [Sar23] Efficient key search (by implementation), unknown amount of security degradation
  - [ZWYGC23] Guess & determine + linearization attack, giving up to 6-bit security degradation

- Mainly, there are two vulnerabilities in the structure of AIM
  - Low degree representation in n variables ⇒ Fast exhaustive search attack
  - Common input to the parallel Mersenne S-boxes ⇒ Structural vulnerability

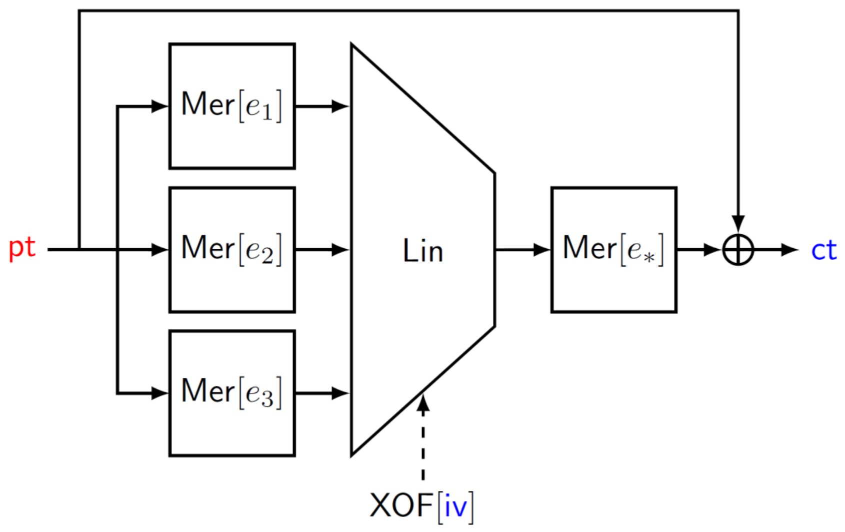
[LMOM23] F. Liu, M. Mahzoun, M. Øygarden, and W. Meier. *Algebraic Attacks on RAIN and AIM Using Equivalent Representations*. IACR Cryptology ePrint Archive. Report 2023/1133. https://eprint.iacr.org/2023/1133.
[Liu23] F. Liu. *Mind Multiple Power Maps: Algebraic Cryptanalysis of Full AIM for Post-quantum Signature Scheme AIMer*. In private communication. 2023.
[Sar23] M. O. Saarinen. *Round 1 (Additional Signatures) OFFICIAL COMMENT: AIMer*. https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/BI2iIXblNy0.
[ZWYGC23] K. Zhang, Q. Wang, Y. Yu, C. Guo, and H. Cui. *Algebraic Attacks on Round-Reduced RAIN and Full AIM-III*. IACR Cryptology ePrint Archive. Report 2023/1397. https://eprint.iacr.org/2023/1397. To appear Asiacrypt 2023.

Fast Exhaustive Search Attack



$$\mathrm{AIM[iv](pt) = ct}$$
$$\Leftrightarrow F(x) = y \;\&\; \deg F = d$$

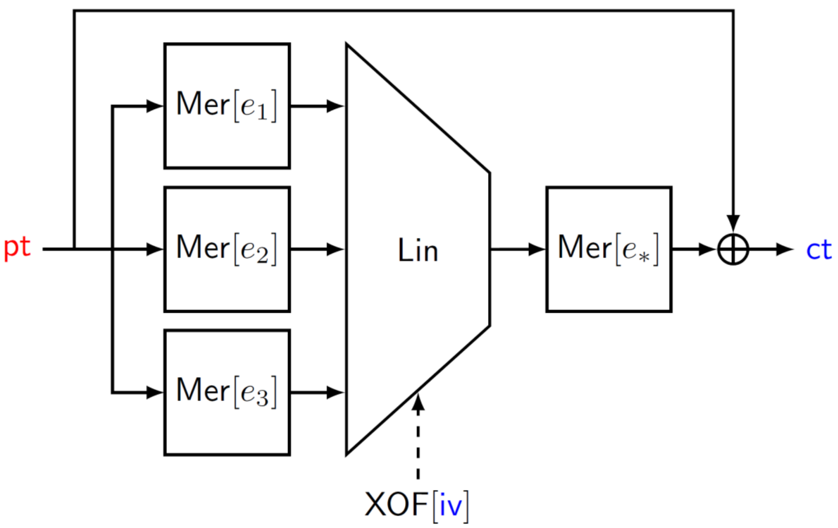
Fast Exhaustive Search Attack



$$\mathrm{AIM[iv](pt)} = \mathrm{ct}$$
$$\Leftrightarrow F(x) = y \;\&\; \deg F = d$$

- Boolean polynomial system can be brute-force searched with $4d \log n \, 2^n$ computation and $O(n^{d+2})$

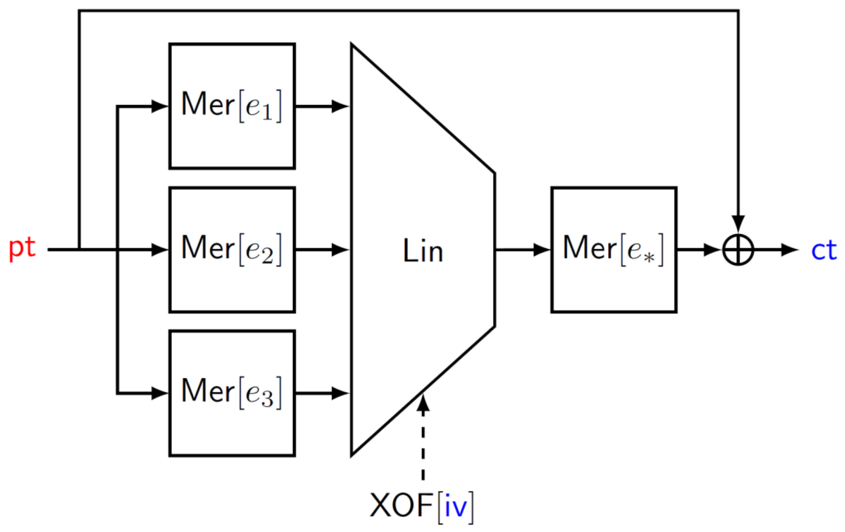
Fast Exhaustive Search Attack



$$\mathrm{AIM[iv](pt)} = \mathrm{ct}$$
$$\Leftrightarrow F(x) = y \;\&\; \deg F = d$$

- Boolean polynomial system can be brute-force searched with $4d \log n \, 2^n$ computation and $O(n^{d+2})$
- If degree d is small enough, this type of fast exhaustive search may be faster than naive brute-force search

Fast Exhaustive Search Attack



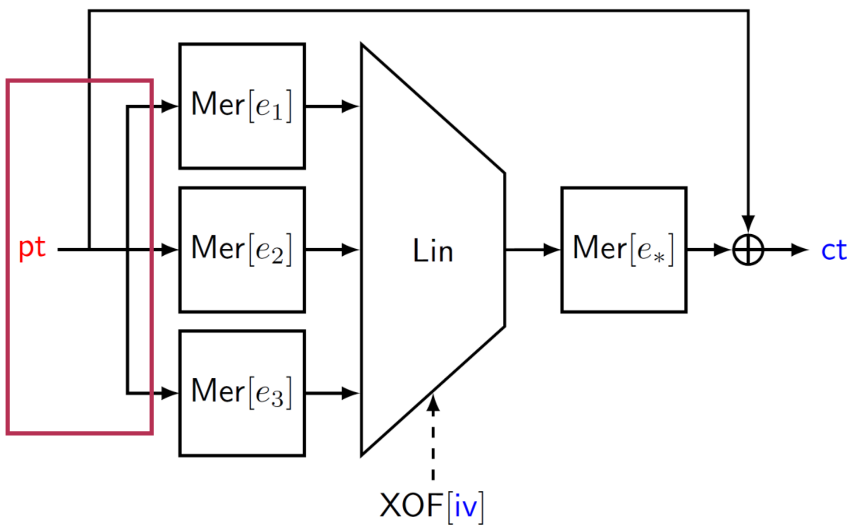
$$\mathrm{AIM[iv](pt)} = \mathrm{ct}$$
$$\Leftrightarrow F(x) = y \;\&\; \deg F = d$$

- Boolean polynomial system can be brute-force searched with $4d \log n\, 2^n$ computation and $O(n^{d+2})$
- If degree d is small enough, this type of fast exhaustive search may be faster than naive brute-force search
- The result of Liu et al. [LMOM23]

	n	Deg	Log(Time) [bits]	Log(Mem) [bits]
AIM-I	128	10	136.2 (−10.2)	61.7
AIM-III	192	14	200.7 (−11.2)	84.3
AIM-V	256	15	265.0 (−12.0)	95.1

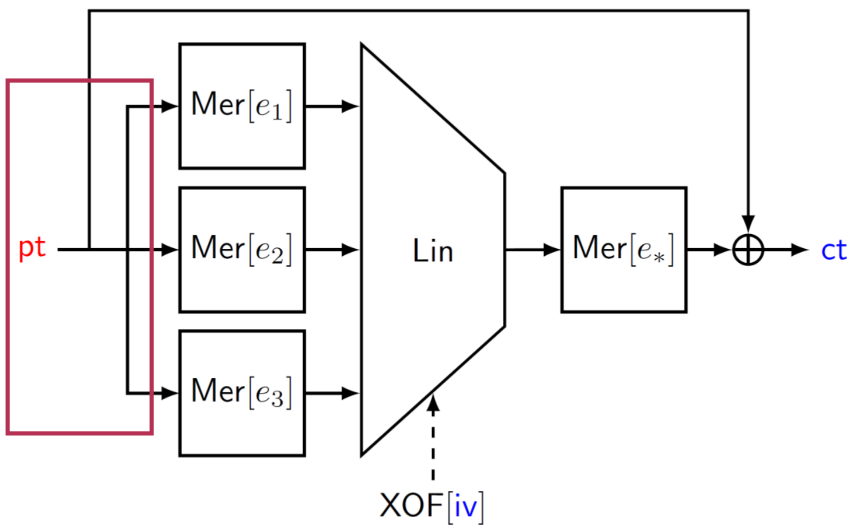
* Compared to the claimed security level

Structural Vulnerability



Inputs to parallel S-boxes are all the same

Structural Vulnerability

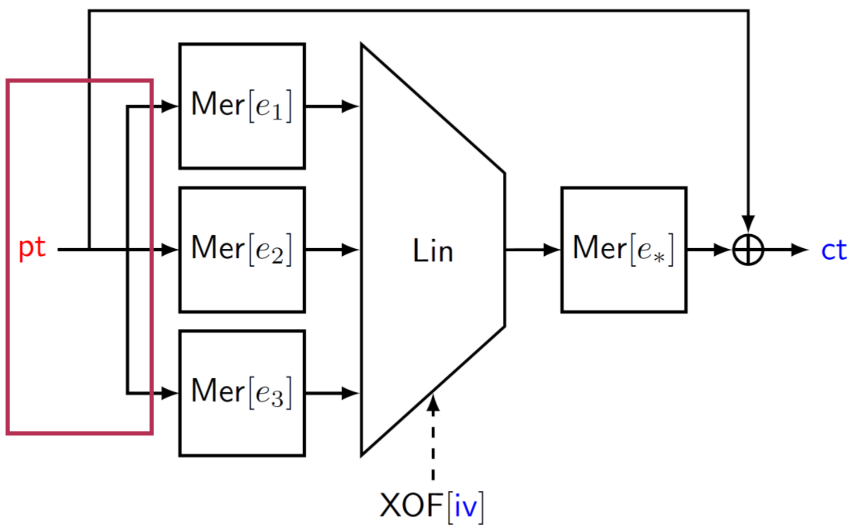


Inputs to parallel S-boxes are all the same

- Find some $d|(2^n - 1)$ such that

$$\begin{cases} \mathrm{Mer}[e_1](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_1} \cdot \mathrm{pt}^{2^{t_1}} \\ \mathrm{Mer}[e_2](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_2} \cdot \mathrm{pt}^{2^{t_2}} \\ \mathrm{Mer}[e_3](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_3} \cdot \mathrm{pt}^{2^{t_3}} \end{cases}$$

Structural Vulnerability



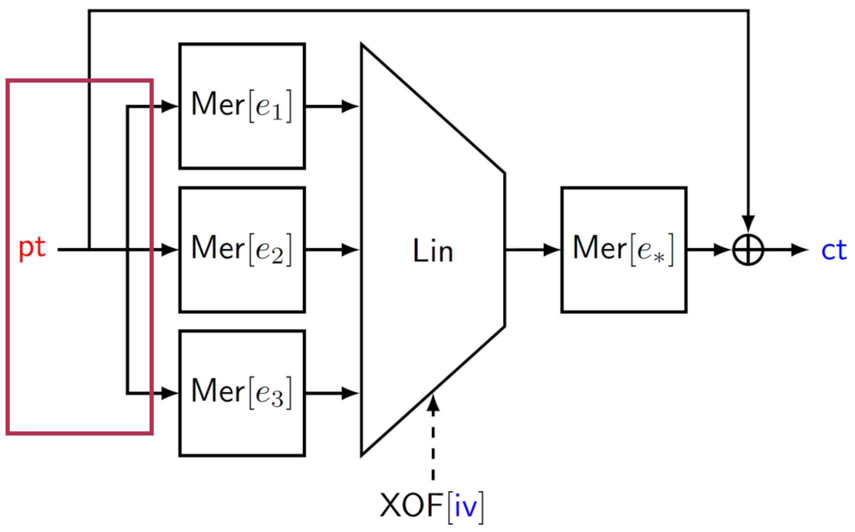
Inputs to parallel S-boxes are all the same

- Find some $d|(2^n - 1)$ such that

$$\begin{cases} \mathrm{Mer}[e_1](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_1} \cdot \mathrm{pt}^{2^{t_1}} \\ \mathrm{Mer}[e_2](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_2} \cdot \mathrm{pt}^{2^{t_2}} \\ \mathrm{Mer}[e_3](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_3} \cdot \mathrm{pt}^{2^{t_3}} \end{cases}$$

- When pt^d is guessed, above system becomes linear

Structural Vulnerability



Inputs to parallel S-boxes are all the same

- Find some $d|(2^n - 1)$ such that

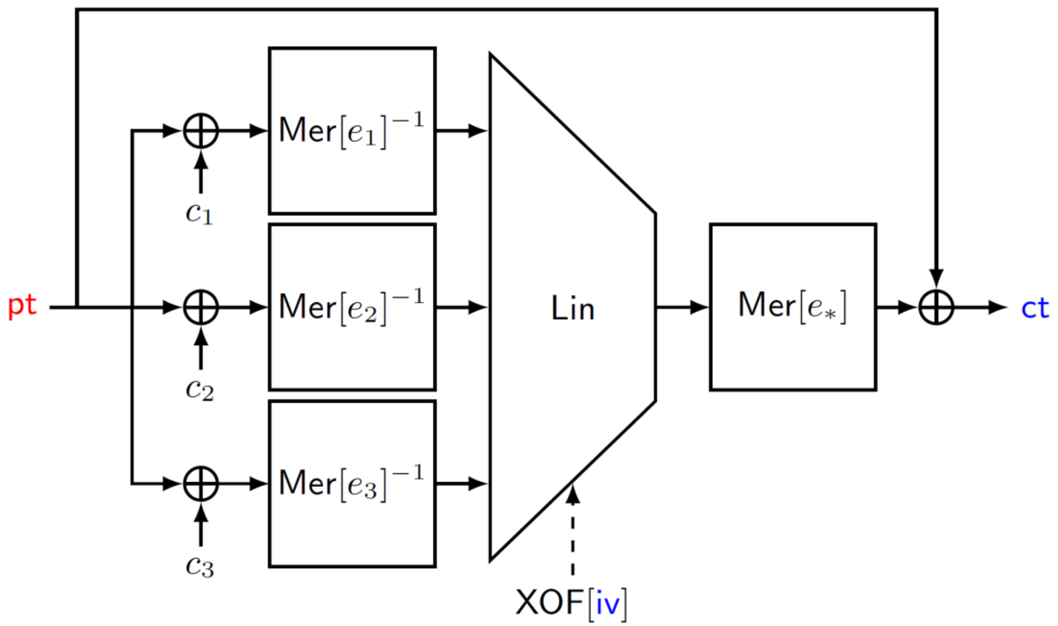
$$\begin{cases} \mathrm{Mer}[e_1](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_1} \cdot \mathrm{pt}^{2^{t_1}} \\ \mathrm{Mer}[e_2](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_2} \cdot \mathrm{pt}^{2^{t_2}} \\ \mathrm{Mer}[e_3](\mathrm{pt}) = \left(\mathrm{pt}^d\right)^{s_3} \cdot \mathrm{pt}^{2^{t_3}} \end{cases}$$

- When pt^d is guessed, above system becomes linear
- The result of Zhang et al. [ZWYGC23]

	n	d	Log(Time) [enc]
AIM-I	128	5	125.7 (-2.3)
AIM-III	192	45	186.5 (-5.5)
AIM-V	256	3	254.4 (-1.6)

* Compared to the claimed security level

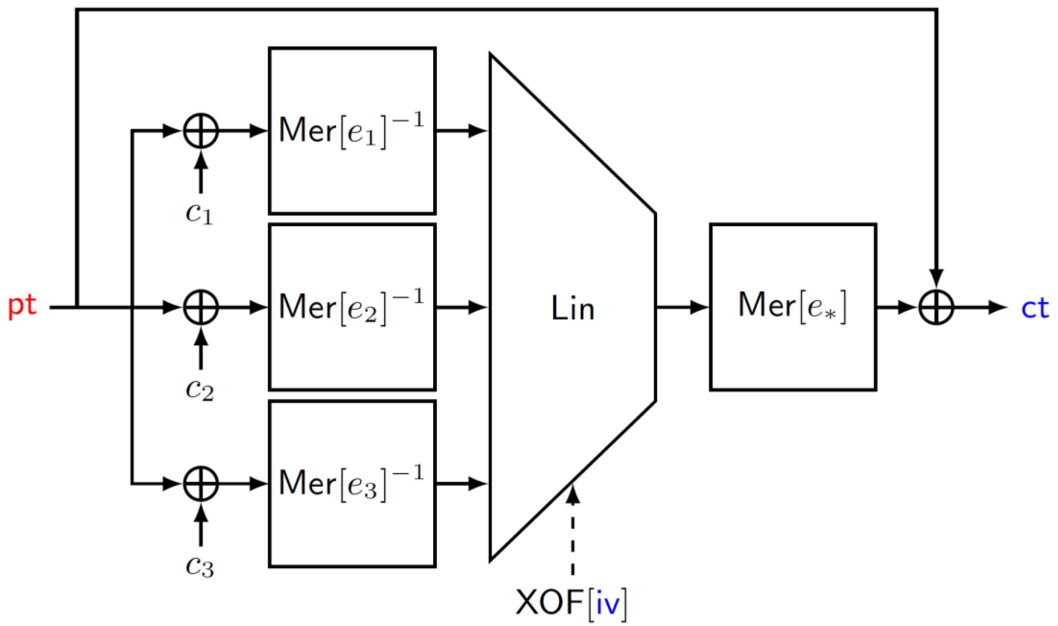
AIM2: Secure Patch for Algebraic Attacks



- 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

* S. Kim et al. "Mitigation on the AIM Cryptanalysis". Cryptology ePrint Archive. Report 2023/1474

AIM2: Secure Patch for Algebraic Attacks

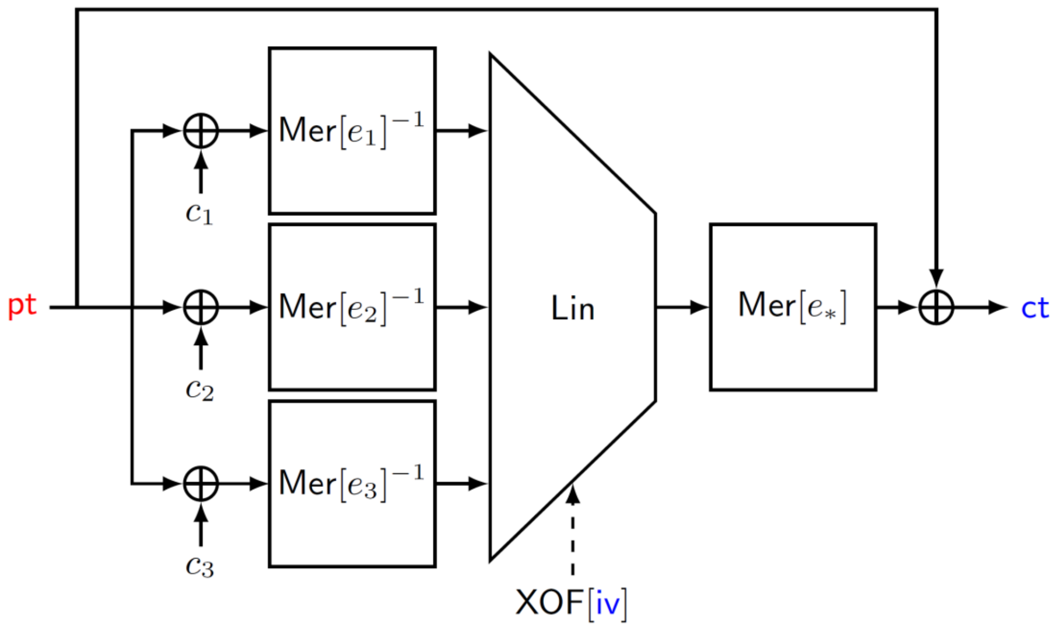


- Inverse Mersenne S-box
  - $\mathrm{Mer}[e]^{-1}(x) = x^a$
  - $a = (2^e - 1)^{-1} \bmod (2^n - 1)$
  - More resistant to algebraic attacks

- Larger exponents
  - To mitigate fast exhaustive search

* S. Kim et al. "Mitigation on the AIM Cryptanalysis". Cryptology ePrint Archive. Report 2023/1474

AIM2: Secure Patch for Algebraic Attacks

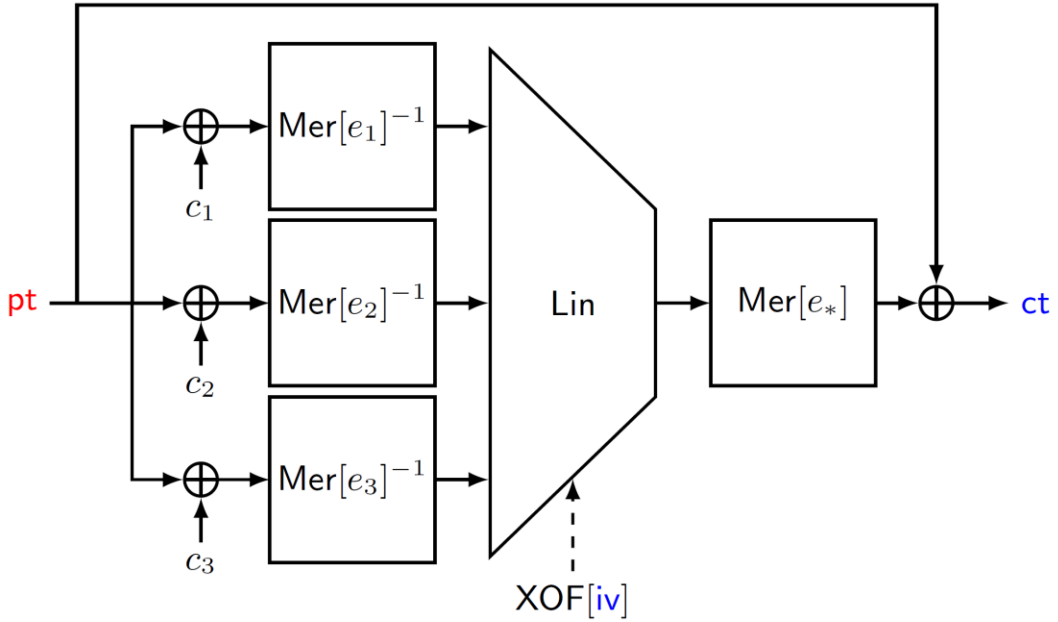


- Inverse Mersenne S-box
  - $\mathrm{Mer}[e]^{-1}(x) = x^a$
  - $a = (2^e - 1)^{-1} \bmod (2^n - 1)$
  - More resistant to algebraic attacks
- Larger exponents
  - To mitigate fast exhaustive search
- Fixed constant addition
  - To differentiate inputs of S-boxes
  - Increase the degree of composite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

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

* S. Kim et al. "Mitigation on the AIM Cryptanalysis". Cryptology ePrint Archive. Report 2023/1474

AIM2: Secure Patch for Algebraic Attacks



Scheme	λ	n	ℓ	e_1	e_2	e_3	e_*
AIM2-I	128	128	2	49	91	-	3
AIM2-III	192	192	2	17	47	-	5
AIM2-V	256	256	3	11	141	7	3

- 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

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

* S. Kim et al. "Mitigation on the AIM Cryptanalysis". Cryptology ePrint Archive. Report 2023/1474

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
  - 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
  - LC/DC: almost same
  - Quantum attacks: complexities change not critically
- Performance
  - Signature size: exactly the same
  - Sign/verify time: $\leq$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
  - LC/DC: almost same
  - Quantum attacks: complexities change not critically
- Performance
  - Signature size: exactly the same
  - Sign/verify time: $\leq$10% increase
- White paper can be found in our website and ePrint Archive 2023/1474

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	32	5904	0.59	0.53
AIMer-L1	32	4176	4.42	4.31
AIMer2-L1	32	5904	0.61	0.53
AIMer2-L1	32	4176	4.47	4.33

* Performance figures of AIMer has been updated from the proceeding version

Conclusion

- Summary
  - We propose symmetric primitive AIM, which is efficiently provable in BN++ proof system
  - AIM has recently been analyzed up to 12-bit security degradation
  - We patched AIM to mitigate the analyses (AIM2) without significant performance overhead
  - The document about AIM2 can be found in ePrint Archive 2023/1474

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

Thank you!
Check out our website!