



Algebraic Attacks on Round-Reduced RAIN and Full AIM-III

Kaiyi Zhang¹(✉) , Qingju Wang² , Yu Yu^{1,3} , Chun Guo^{4,5,6} ,
and Hongrui Cui¹ 

¹ Shanghai Jiao Tong University, Shanghai, China
{kzoacn,yyuu,rickfreeman}@sjtu.edu.cn

² Telecom Paris, Institut Polytechnique de Paris, Palaiseau, France
qingju.wang@telecom-paris.fr

³ Shanghai Qi Zhi Institute, Shanghai, China

⁴ School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China

⁵ Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Qingdao, Shandong 266237, China

⁶ Shandong Research Institute of Industrial Technology, Jinan, Shandong 250102, China

Abstract. Picnic is a NIST PQC Round 3 Alternate signature candidate that builds upon symmetric primitives following the MPC-in-the-head paradigm. Recently, researchers have been exploring more secure/efficient signature schemes from conservative one-way functions based on AES, or new low-complexity one-way functions like RAIN (CCS 2022) and AIM (CCS 2023 and Round 1 Additional Signatures announced by NIST PQC). The signature schemes based on RAIN and AIM are currently the most efficient among MPC-in-the-head-based schemes, making them promising post-quantum digital signature candidates.

However, the exact hardness of these new one-way functions deserves further study and scrutiny. This work presents algebraic attacks on RAIN and AIM for certain instances, where one-round RAIN can be compromised in $2^{n/2}$ for security parameter $n \in \{128, 192, 256\}$, and two-round RAIN can be broken in $2^{120.3}$, $2^{180.4}$, and $2^{243.1}$ encryptions, respectively. Additionally, we demonstrate an attack on AIM-III (which aims at 192-bit security) with a complexity of $2^{186.5}$ encryptions. These attacks exploit the algebraic structure of the power function over fields with characteristic 2, which provides potential insights into the algebraic structures of some symmetric primitives and thus might be of independent interest.

Keywords: Algebraic Attacks · Power Mapping · Arithmetization Oriented Primitives · RAIN · AIM

1 Introduction

With significant advancements in quantum computing over the past decades, the security threats of quantum computers are increasingly becoming a reality. As a

response, the cryptographic community is seeking post-quantum alternatives to the widely deployed public-key cryptography algorithms, the most noteworthy of which is NIST’s post-quantum cryptography (PQC) standardization process.¹ This has motivated numerous novel designs as well as analyses of the underlying hardness assumptions.

Towards post-quantum digital signatures, a popular approach is to employ the MPC-in-the-Head (MPCitH) paradigm, proposed by Ishai et al. [24]. In detail, MPCitH provides a general construction of a zero-knowledge proof for an NP relation by making a black-box use of any secure multi-party computation protocol for a related functionality. A major application of the MPCitH paradigm is to construct post-quantum digital signatures, which are essentially a non-interactive zero-knowledge proof of knowledge (NIZKPoK) that the input of a specific one-way function (secret key of the signature) corresponds to the one-way function’s output (public key of the signature). Note that the message to be signed is involved in the challenge generation of NIZKPoK. The NIZKPoK, when based on quantum-resistant one-way functions, gives rise to promising candidates for post-quantum signatures. Chase et al. [7, 25] pioneered NIZKPoK-based signatures, and they designed the Picnic scheme that advanced to the third round of the NIST PQC standardization process.

Subsequent improvements to MPCitH-based signatures follow two approaches. The first approach [6, 14, 15] sticks to standard primitives such as the AES, and focuses on improving the efficiency of the zero-knowledge proof. Concretely, such designs instantiate the block cipher-based OWF $H(k) = E_k(P)$ (with P a public constant) with AES.² Since the performance of MPCitH-based signatures is typically closely tied to the number of non-linear operations in the circuit of the underlying one-way function, the large circuit size of AES constitutes the bottleneck. Generally, symmetric primitives, which aim to minimize the cost related to the number of non-linear operations, are called *arithmetization-oriented* symmetric primitives. With this in mind, the second approach is devoted to designing efficient arithmetization-oriented block ciphers for $H(k) = E_k(P)$ or even new one-way functions. In fact, Picnic already employed this idea and instantiated $H(k) = E_k(P)$ with an MPC-friendly block cipher LowMC, which aims to minimize the number of AND gates over \mathbb{F}_2 . The idea was further extended by Dobraunig et al. [20] and Kim et al. [26]. Dobraunig et al. proposed a novel block cipher to instantiate $H(k) = E_k(P)$. More detailedly, RAIN may be viewed as an iterated Even-Mansour scheme with round permutation defined upon inversions in \mathbb{F}_{2^n} and random matrices in $\mathbb{F}_2^{n \times n}$, where n is the block size. Such round permutations may also be viewed as generalizations of the AES S-box. Kim et al. [26] made a step further and proposed a “tweakable” OWF AIM that is built upon a novel construction and employs the Mersenne power function as its S-box. Both RAIN and AIM maintained to reduce the number of multiplications over \mathbb{F}_{2^n} , and they enjoy the shortest

¹ <https://csrc.nist.gov/projects/post-quantum-cryptography>.

² The one-wayness of $H(k) = E_k(P)$ is equivalent to the key recovery security of E with a single plaintext/ciphertext pair: see [20] for a proof.

signature size as well as comparable signing/verification time among MPCitH-based signatures (to the best of our knowledge). In July 2023, AIMer has been submitted as one of the NIST PQC Round 1 Additional Signatures in the category of Symmetric-based Signatures.³

We remark that the block cipher RAIN is only intended to be used in the OWF construction $H_k = E_k(P)$, the security of which is equivalent to the key recovery security of E with a single plaintext/ciphertext pair. Therefore, classical attacks involving multiple data, including statistical attacks like the differential attack, are not immediately relevant in this setting.

The MPC-friendly primitive-based approach proves a huge success w.r.t. performance: all of the aforementioned designs, i.e., (LowMC-based) Picnic, RAIN-based RAINIER and AIM-based AIMer, managed to reduce multiplications (over \mathbb{F}_2 or \mathbb{F}_{2^n}) as well as the signature size (as mentioned). On the other hand, classical cryptanalytic methods (e.g., differential and linear attacks) are mostly inapplicable due to the limited available data. These have motivated investigating dedicated cryptanalytic methods and deepening the understanding of such designs. For example, since proposed, LowMC has undergone quite some cryptanalysis [5, 18, 19, 28, 30, 32, 38]. In this work we focus on RAIN and AIM and design novel algebraic attacks breaking the one-wayness of certain instances, as elaborated below.

1.1 Our Contributions

Here we give an overview of our attacks and propose possible countermeasures.

Overview of the New Algebraic Attacks. RAIN employs the same kind of S-box as the AES, which is the multiplicative inverse function over \mathbb{F}_{2^n} (with zero mapped to zero). Our initial focus was to linearize this S-box. We progress in this direction by discovering the following fact:

$$x^{254} = (x^{17})^{14} \cdot x^{16}$$

where x^{254} is the non-linear layer of AES and the formula is over \mathbb{F}_{2^8} . For the two terms of the formula:

- In the first term, 17 is a divisor of 255, meaning that x^{17} has only $255/17 = 15$ possible choices for $x \in \mathbb{F}_{2^8} \setminus \{0\}$.
- In the second term, the square function over \mathbb{F}_{2^n} is linear, which means that square function x^2 can be represented by matrix multiplication as $x^2 = Mx$ for an invertible matrix $M \in \mathbb{F}_2^{8 \times 8}$. Hence, the mapping $x \mapsto x^{16}$ can be expressed as $x \mapsto M'x$ for $M' = M^4$.

Therefore, if we guess the value of $x^{17} = \alpha$ from 15 possible choices, we can express x^{254} as $x^{254} = \alpha^{14} \cdot x^{16} = M''x$ for some M'' . This linearizes the non-linear layer of AES, but unfortunately, no attacks better than the current state-of-the-art has been found based on this fact.

³ <https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures>.

However, RAIN also uses the multiplicative inverse function over \mathbb{F}_{2^n} as its non-linear layer. For an even n , we decompose it similarly to the AES as

$$x^{2^n-2} = (x^{2^{n/2}+1})^{2^{n/2}-2} \cdot x^{2^{n/2}}.$$

As can be seen, $d = 2^{n/2} + 1$ is a divisor of $2^n - 1$. In other words, $\{x^d : x \in \mathbb{F}_{2^n} \setminus \{0\}\}$ is a subgroup of the multiplicative group $\mathbb{F}_{2^n}^*$ of the finite field \mathbb{F}_{2^n} . Therefore x^d has only $|\mathbb{F}_{2^n}^*|/d = (2^n - 1)/(2^{n/2} + 1) = 2^{n/2} - 1$ possible choices. Secondly, $x \mapsto x^{2^{n/2}}$ is a linear function over \mathbb{F}_2 . Guessing at most $2^{n/2} - 1$ possible choices in a similar way, allows the adversary to linearize the non-linear layer. Using this method, we can recover the key of one-round RAIN in $2^{n/2}$ encryptions.

When we move to two-round RAIN, this method will not work directly. Inspired by [30], we introduce quadratic equations into our attack, and propose a new method of deriving quadratic equations from two S-boxes in the form of power mappings. More specifically, in the first step, by linearizing the S-box in the first non-linear layer, we obtain independent linear equations, and free variables for the secret key bits. In the next step, we construct quadratic equations from the second S-box. By substituting unknown variables with free variables obtained in the previous step, we get a sufficient number of linearly independent quadratic equations so that all free variables can be solved efficiently.

Our Attacks on RAIN and AIM. We demonstrate that one-round RAIN can be compromised in $2^{n/2}$ number of encryptions, for $n \in \{128, 192, 256\}$, while two-round RAIN can be broken in $2^{120.3}$, $2^{180.4}$, and $2^{243.1}$ encryptions, respectively. Furthermore, we also show an attack on full AIM-III (which aims at 192-bit security) with $2^{186.5}$ encryptions. Notably, all these attacks are conducted in the one-way function setting, where the adversary has access to only one plaintext/ciphertext pair or one ciphertext in the case of AIM. Besides, the memory cost of our attacks is negligible. Finally, we implement attacks on two-round RAIN and AIM-III, practically showing that there are sufficiently enough linearly independent equations to solve unknown variables. Our implementation can be found at

<https://github.com/kzoacn/LargeSbox/>.

At present, these attacks do not affect the security of the signature scheme which uses three- or more-round RAIN. Moreover, the security of AIM-I (which aims at 128-bit security) and AIM-V (which aims at 256-bit security) are not significantly affected by our attack.

Our attack exploits the algebraic structure of the power function over the binary Galois field, which is widely used in the design of symmetric primitives suitable for MPCitH-based signature schemes. Therefore, our findings provide new insights into the algebraic structures of these primitives, which might be of independent interest.

Restoring the Security of AIM. Lastly, in Sect. 5.2 we discuss some countermeasures that allow AIM to regain security. These include countermeasures

avoiding the simultaneous linearization of two S-boxes, restricting of the order n of extension field \mathbb{F}_{2^n} to be odd, and increasing the number of rounds.

1.2 Related Work

Post-quantum signatures can be built upon other hardness assumptions as well, including lattice assumptions (e.g., CRYSTALS-Dilithium [33] and Falcon [37]), the intractability of Multivariate Quadratic (MQ) problems, etc. Some representative schemes such as SPHINCS+ [23] are solely based on cryptographic hash functions (but rely on much stronger assumptions than the one-wayness of the hash functions). We refer to [39] for a survey.

Since RAIN is an instance of the iterated Even-Mansour scheme, we refer to [13, 16, 17, 21, 36] for generic key recovery attacks. Though, with just one plaintext/ciphertext pair, such generic attacks are either inapplicable (on two rounds) or less efficient than ours (on one round).

During the submission of this work, Liu and Mahzoun proposed attacks on 2-round RAIN and full-round AIM [29]. Since the natural isomorphism between \mathbb{F}_2^n and \mathbb{F}_{2^n} , both RAIN and AIM defined over \mathbb{F}_{2^n} , can be represented as relatively low-degree polynomials over \mathbb{F}_2^n . They proposed highly refined techniques to solve systems of low-degree equations, while we more focus on the essential property of power mapping S-boxes in this work.

1.3 Paper Organization

In Sect. 2, we define notations and give background information for the linearization techniques used in this paper. We present our first attack on one-round RAIN in Sect. 3. In order to attack more rounds of RAIN, we first propose a general algebraic attack framework at the beginning of Sect. 4. Under this framework, we achieve attacks on two-round RAIN and the full rounds of AIM-III. We discuss some countermeasures that allow AIM to reestablish security in Sect. 5. We conclude this paper in Sect. 6.

2 Preliminaries

2.1 Notations

Let p be a prime number, and $q = p^n$ for a positive integer n . Let \mathbb{F}_p denote a finite field with p elements, and let $\mathbb{F}_q = \mathbb{F}_{p^n}$ denote a finite field with characteristic p . The multiplicative group of \mathbb{F}_q is denoted by $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$. We use $[m]$ to denote the set $\{0, 1, \dots, m-1\}$ for $m \in \mathbb{N}^+$. We use n as both the security parameter and the block size, since for both RAIN and AIM the block size equals the security parameter. The size of a set S is denoted by $|S|$. The n -dim identity matrix is denoted by I .

2.2 Finite Field

Theorem 1 (e.g., Theorem 2.1.37 from [34]). *The multiplicative group \mathbb{F}_q^* of a finite field \mathbb{F}_q is cyclic.*

Definition 1. *A generator g of a multiplicative group \mathbb{F}_q^* is called a primitive element of the field \mathbb{F}_q . It is denoted as $\mathbb{F}_q^* = \langle g \rangle$.*

Theorem 2. *Let d be a divisor of $|\mathbb{F}_q^*| = p^n - 1$, and let $X \stackrel{\text{def}}{=}} \{x^d : x \in \mathbb{F}_q^*\}$. The size of X is $|X| = |\mathbb{F}_q^*|/d = (p^n - 1)/d$.*

Theorem 2 tells us that for any appropriate $d > 1$ with $d|(p^n - 1)$, the power function $x \mapsto x^d$ maps inputs from \mathbb{F}_q^* to a proper subset of \mathbb{F}_q^* .

2.3 Linearized Polynomial

Definition 2 (linearized polynomial). *A linearized polynomial is defined as $L(x) \stackrel{\text{def}}{=} \sum_{i \in [n]} a_i x^{p^i}$, where $a_i \in \mathbb{F}_{p^n}$ for some prime p .*

Let $L(x)$ be a linearized polynomial over a finite field \mathbb{F}_p . The map $x \mapsto L(x)$ is a linear map over \mathbb{F}_p , i.e., for all $a, b \in \mathbb{F}_{p^n}$ and $c \in \mathbb{F}_p$, we have $L(a + b) = L(a) + L(b)$ and $c \cdot L(a) = L(c \cdot a)$. Due to the existence of a vector space isomorphism $\mathbb{F}_p^n \cong \mathbb{F}_{p^n}$, we can naturally view an element $x \in \mathbb{F}_{p^n}$ as a vector $\hat{x} \in \mathbb{F}_p^n$. More specifically, there must exist a matrix $M \in \mathbb{F}_p^{n \times n}$ such that $L(x) = M\hat{x}$. For simplicity, we interchangeably view x as an element of \mathbb{F}_{p^n} or a vector $x \in \mathbb{F}_p^n$, so that we can write $L(x) = Mx$.

In the case of the bijective mapping $\mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n} : x \mapsto x^{2^k}$, we can represent it as $\mathbb{F}_2^n \rightarrow \mathbb{F}_2^n : x \mapsto Mx$ for some invertible binary matrix M .

2.4 Linearization and Multivariate Quadratic Equations over Finite Field

Linearization Attacks. Solving multivariate equations is regarded as an NP-hard problem in general. However, in some special cases, multivariate polynomial systems of equations over finite fields can be solved in polynomial time (see e.g., [27]). The core idea of linearization is to turn a system of non-linear equations into a linear system by treating each monomial as a separate variable. In general, the method generates polynomials of some degree, up to the point where the number of equations exceeds the number of monomials so a solution can be obtained by some linear algebra. In symmetric cryptography, it is usually assumed that the attackers have access to sufficiently many equations to linearize the system. Recall that the number of possible monomials in a degree d polynomial in $\mathbb{F}[x_1, \dots, x_n]$, where $|\mathbb{F}| > d$, is $\mathcal{B}_{n,d} = \binom{n+d}{d}$. The linearization attack requires $\mathcal{O}(\mathcal{B}_{n,d}^\omega)$ multiplications in \mathbb{F} , where $2 < \omega \leq 3$, $\mathcal{O}(\mathcal{B}_{n,d})$ data complexity, and $\mathcal{O}(\mathcal{B}_{n,d}^2)$ memory complexity.

Multivariate Quadratic Equations Over \mathbb{F}_2 . In this paper, the main focus is the quadratic multivariate equations. A set of multivariate quadratic equations consists of equations of the form

$$\sum_{i,j \in [n]} a_{i,j}x_i x_j + \sum_{i \in [n]} b_i x_i + c = 0,$$

where $a_{i,j}, b_i, c \in \mathbb{F}_2$ are constants, and x_1, x_2, \dots, x_n are the unknown variables. When applying the linearization technique to the quadratic case, it simply involves replacing each quadratic term with a new separate variable. Suppose we want to replace $x_i x_j$ with a new variable $y_k = x_i x_j$. Note that in \mathbb{F}_2 , $x_i^2 = x_i$ for any i always holds. Therefore the only possible quadratic terms are of the form $y_k = x_i x_j$ with $i < j$, which lead to $n(n - 1)/2$ instead of $n(n + 1)/2$ new variables. Therefore, combined with the n variables x_i 's, the total number of variables will be $m = n + n(n - 1)/2$.

While there are more sophisticated methods available, such as relinearization [27] or XL [11], the linearization technique suffices for our attacks.

3 Preliminary Algebraic Attack on One-Round RAIN

In this section, we present the algebraic attack on one-round RAIN. We first start with the specification of the target cipher.

3.1 Description of RAIN

RAIN is the one-way function designed for the signature scheme Rainier. RAIN is denoted as a keyed permutation $E_k(P) = C$, where the input P and the output C are public, and the secret key k is private. As a one-way function used in the signature scheme, it is $H(k) = E_k(P)$. The keyed permutation is a concatenation of r round functions R_i . For each round $1 \leq i < r$, the round function R_i is defined by

$$R_i(x) = M_i(S(x + k + c^{(i)}))$$

and the last round function is

$$R_r(x) = k + S(x + k + c^{(r)}),$$

where $c^{(i)} \in \mathbb{F}_2^n$ is the round constant, and $M_i \in \mathbb{F}_2^{n \times n}$ is the linear layer matrix used in the i -th round. Each M_i can be represented as a linearized polynomial over \mathbb{F}_{2^n} as defined in Definition 2. The details for the generation of the round constants and matrices are referred to in the design paper [20]. The non-linear layer of RAIN is defined as $S : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ such that

$$S(x) = x^{2^n - 2} = \begin{cases} x^{-1}, & x \neq 0 \\ 0, & x = 0 \end{cases}$$

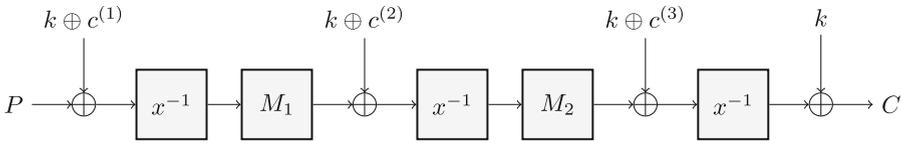


Fig. 1. RAIN permutation with $r = 3$.

A graphical overview of the construction with the round number $r = 3$ is shown in Fig. 1.

Parameter Sets. The block size of RAIN is $n \in \{128, 192, 256\}$. Two variants for each block size are recommended with the round number $r = 3, 4$, which are used in the signature Rainier_{r-n} . We denote the one-way function with the block size n as $\text{RAIN-}n$ in our presentation, neglecting the number of rounds r .

The finite field and irreducible polynomials of RAIN are defined as follows:

- For 128-bit security, the finite field is $\mathbb{F}_{2^{128}}$ with irreducible polynomial $X^{128} + X^7 + X^2 + X + 1$.
- For 192-bit security, the finite field is $\mathbb{F}_{2^{192}}$ with irreducible polynomial $X^{192} + X^7 + X^2 + X + 1$.
- For 256-bit security, the finite field is $\mathbb{F}_{2^{256}}$ with irreducible polynomial $X^{256} + X^{10} + X^5 + X^2 + 1$.

3.2 Attack on One-Round RAIN

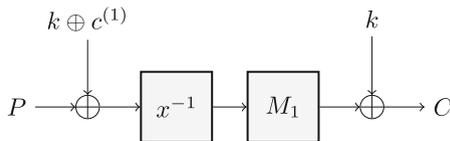


Fig. 2. RAIN permutation with $r = 1$.

In this subsection, we describe our attacks on one-round RAIN (Fig. 2). Here, one round of RAIN means a regular round, i.e., it consists of the nonlinear operation S-box, and the linear operation M_1 . The relation of the public input P and public output C , and the secret key k can be expressed as

$$M_1(P + k + c^{(1)})^{-1} + k = C. \tag{1}$$

To recover the key k is to find the root of Eq. (1).

Before giving the procedures of our attack, we first explain some observations about the multiplicative inverse function (S-box) that the attack relies on.

Proposition 1. For an even n , the multiplicative inverse function x^{-1} over \mathbb{F}_{2^n} can be linearized by guessing $2^{n/2} - 1$ possible values for x^d , where $d = 2^{n/2} + 1$.

Proof. For an even n , the multiplicative inverse function over \mathbb{F}_{2^n} can be represented as

$$x^{2^n-2} = (x^{2^{n/2}+1})^{2^{n/2}-2} \cdot x^{2^{n/2}}. \tag{2}$$

Let $d = 2^{n/2} + 1$, $s = 2^{n/2} - 2$, and $t = n/2$, Eq. (2) can be simplified as

$$x^{2^n-2} = (x^d)^s \cdot x^{2^t}. \tag{3}$$

We proceed to performing an individual analysis of both terms within the simplified expression in Eq. (3) as follows:

- i) According to Theorem 2, d is a divisor of $2^n - 1$, x^d takes only $(2^n - 1)/d = 2^{n/2} - 1$ possible values.
- ii) According to Sect. 2.3, $x \mapsto x^{2^t}$ is equivalent to $x \mapsto Mx$ for an invertible matrix M .

It means that we can linearize the multiplicative inverse function over \mathbb{F}_{2^n} (for even n) by guessing $2^{n/2} - 1$ possible values for x^d . ■

Based on Proposition 1, we give the detailed steps of our attack as follows:

1. Enumerate an element α from the set $D = \{x^d : x \in \mathbb{F}_{2^n}^*\}$, then let the value of $(P + k + c^{(1)})^d$ be α .
2. Compute $(P + k + c^{(1)})^{-1} = \alpha^s \cdot (P + k + c^{(1)})^{2^t} = Mk + b$ for a matrix $M \in \mathbb{F}_2^{n \times n}$ and a vector $b \in \mathbb{F}_2^n$.
3. Substitute $(P + k + c^{(1)})^{-1}$ in Eq. (1) with $Mk + b$ and obtain $M'k + b' = 0$, where $M' = M_1M + I$ and $b' = M_1b + C$.
4. Use Gaussian elimination to solve $M'k + b' = 0$ and obtain k^* as the solution for k .
5. Check if k^* is the correct key by checking if $R_1(k^*, P) + k^* = C$. If not, we repeat Step 1.

Enumerating α . Let g be a generator of $\mathbb{F}_{2^n}^*$. According to Theorem 1, g generates the entire $\mathbb{F}_{2^n}^*$, i.e., any element $x \in \mathbb{F}_{2^n}^*$ can be represented as $x = g^i$ for $0 \leq i \leq 2^n - 1$. It follows that $D = \{x^d : x \in \mathbb{F}_{2^n}^*\} = \{g^{i \cdot d} : i = 0, 1, \dots, (2^n - 1)/d - 1\}$. Therefore, we can generate the desired set D by enumerating $i \in \{0, 1, \dots, (2^n - 1)/d - 1\}$ and compute $g^{i \cdot d}$.

Complexity Analysis. There are $(|\mathbb{F}_{2^n}| - 1)/d$ possible choices of α . For each guess, it takes $\mathcal{O}(n^3)$ time to solve the linear equations. Therefore, the time complexity of this attack is $T_{\text{bit}} = (2^n - 1)/d \cdot n^3$ in terms of bit operations. To convert the time complexity in a number of encryptions, we re-calculate it as $T = T_{\text{bit}}/n^3$. This is a useful conversion because it allows us to directly compare the complexity of this attack to that of a brute force attack, which has a complexity of exactly 2^n . The memory complexity of our approach is negligible since we only need to store the equations.

Table 1. Results for one-round RAIN. n is the block size and the security parameter, r is the number of rounds attacked, d is the divisor we use to determine the guess of $\alpha (= x^d)$, t determines the linear term in the decomposition of $2^n - 2$, and T is the time complexity of the attack.

Scheme	n	r	d	t	T
RAIN-128	128	1	$2^{64} + 1$	64	2^{64}
RAIN-192	192	1	$2^{96} + 1$	96	2^{96}
RAIN-256	256	1	$2^{128} + 1$	128	2^{128}

Parameter Sets of Our Attack. The detailed parameters of our attack on one-round RAIN are given in Table 1.

Toy Example. We provide a concrete example for the attack on one-round RAIN, which can be found in Appendix A.1.

4 Algebraic Attack Framework and Its Application to Two-Round RAIN and Full AIM-III

In the attack on one-round RAIN, we use a linearization technique to handle the non-linear S-box. With a carefully chosen divisor d of n , we guess the value of x^d and then decompose x^{-1} as a product of $(x^d)^s$ and x^{2^t} , where the former is a constant and the latter is linear.

In order to attack more rounds of RAIN, our initial attempt was to straightforwardly linearize more than one S-box using this method. We note that the complexity of the linearization of one S-box in such a way is roughly $(|\mathbb{F}_{2^n}| - 1)/d$, so the larger the divisor d is, the smaller the complexity is. However, the largest possible divisor d that can be used to linearize x^{-1} is $2^{n/2} + 1$, therefore the cost of this technique is at least $(|\mathbb{F}_{2^n}| - 1)/d \approx 2^{n/2}$ for each S-box. Consequently, the unfavorable ramification for linearizing more than one S-box is that the overall complexity will not be superior to the brute force. Therefore, we must explore alternative techniques to tackle the two-round RAIN.

4.1 Algebraic Attack Framework

We devote ourselves to the algebraic attacks on more than one round of RAIN, i.e., more than one S-box. The efficiency of algebraic cryptanalysis heavily depends on the number of variables, the number of equations, and the degrees for the system of equations. Numerous efforts have been undertaken to investigate the number of linearly independent quadratic equations obtained from power functions over \mathbb{F}_{2^n} [8, 10, 22, 35]. For instance, it has been pointed out that for the inverse mapping which is the S-box employed in RAIN, this number is $5n$ [8, 10]; for the power mapping with the exponentiation as a Mersenne number such as $x^{2^e - 1}$, this number is $3n$ [35]. The security of RAIN and AIM against

attacks exploiting algebraic properties such as Gröbner basis attack, is carefully evaluated by representing primitives in equations over both \mathbb{F}_2 and \mathbb{F}_{2^n} of the highest degree to ensure the algebraic analysis infeasible.

In this subsection, we further leverage Proposition 1, and propose a new method of deriving quadratic equations from two S-boxes in the form of power mappings. More specifically, in the first step, by linearizing the S-box in the first non-linear layer, we obtain independent linear equations, and free variables for the secret key bits. In the next step, we construct quadratic equations from the second S-box. By substituting unknown variables with free variables obtained in the previous step, we get a sufficient number of linearly independent quadratic equations, enabling us to solve all free variables.

In the following, we describe details on how to construct linear equations and quadratic equations from the S-boxes of power mappings.

Deriving Linear Equations from x^d . Recall that d is one divisor of $2^n - 1$. We observe that for such a d , we can obtain linearly independent equations from $x^d = \alpha$. For example, let the field be \mathbb{F}_{2^8} . If we guess $x^5 = 1$, then we can obtain the equation $x^{16} + x = 0$ because $x^{16} = (x^5)^3 \cdot x = x$. Representing the map $x \mapsto x^{16}$ as a matrix multiplication $x \mapsto Mx$, we can find that 4 out of 8 equations from $Mx + x = 0$ are linearly independent, which can be used to solve for x .

Suppose $d \cdot s_1 + 2^{t_1} = d \cdot s_2 + 2^{t_2}$, by guessing $x^d = \alpha$, we can obtain one linear equation $\alpha^{s_1} \cdot x^{2^{t_1}} + \alpha^{s_2} \cdot x^{2^{t_2}} = 0$ because $(x^d)^{s_1} \cdot x^{2^{t_1}} = (x^d)^{s_2} \cdot x^{2^{t_2}}$. The number of linearly independent equations (denoted by γ) can be found by experiment.

Deriving Quadratic Equations from x^d . We notice that when guessing $x^5 = 1$, we obtain the quadratic equation $x^4 \cdot x + 1 = 0$. More generally, we can obtain quadratic equations from expressions of the form x^d as follows.

Suppose $d \cdot s_1 + 2^{t_1} + 2^{t_2} = d \cdot s_2 + 2^{t_3} + 2^{t_4}$, we have quadratic equations $\alpha^{s_1} \cdot x^{2^{t_1}} \cdot x^{2^{t_2}} + \alpha^{s_2} \cdot x^{2^{t_3}} \cdot x^{2^{t_4}} = 0$ because $(x^d)^{s_1} \cdot x^{2^{t_1}} \cdot x^{2^{t_2}} = (x^d)^{s_2} \cdot x^{2^{t_3}} \cdot x^{2^{t_4}}$.

Deriving Equations for the S-Boxes. The S-boxes in both RAIN and AIM are power functions of the form $S(x) = x^h$. These functions can be leveraged in two ways to exploit their algebraic structures.

The first method involves linearizing x^h by making a guess for x^d and using the expression $x^h = (x^d)^s \cdot x^{2^t}$. The output of the S-box can be denoted as y , and this technique can be used to simplify the algebraic representation of $S(x)$.

The second method involves representing both S-boxes as quadratic equations. Subsequent subsections will provide more details on this approach.

Attack Framework. To summarize, our attacks on RAIN and AIM follow a similar framework, which can be outlined as follows:

1. Guess the value of x^d , where x and d are carefully chosen bases and exponents, respectively.
2. Derive a sufficient number of linear and quadratic equations over \mathbb{F}_2 from x^d and the internal relations of RAIN/AIM.

3. Solve the equations using linearization techniques and Gaussian elimination.

Under the new general algebraic attack framework, we will demonstrate the effectiveness of our attacks by applying it to two-round RAIN and full rounds of AIM-III in the following subsections.

4.2 Attack on Two-Round RAIN

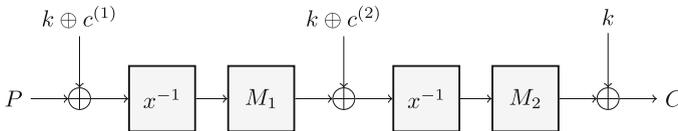


Fig. 3. RAIN permutation with $r = 2$.

The structure of the two-round RAIN is illustrated in Fig. 3. Once more, by two-round RAIN we mean two regular rounds of the RAIN permutation.

For an even value n , an (partially) integer factorization of $2^n - 1$ can be obtained by repeatedly applying the squared difference formula:

$$2^n - 1 = (2^{n/2} + 1) \cdot (2^{n/4} + 1) \cdots (2^{n/2^t} - 1),$$

where t is the maximum i such that 2^i is a divisor of n . In our parameter sets for the attack on two-round RAIN cipher, the divisor d is always of the form $2^w + 1$, where w is a positive integer.

Linearization of the First S-Box. The first step in our attack involves guessing the value of $(P + k + c^{(1)})^d = \alpha$, which linearizes the expression of the first S-box by $(P + k + c^{(1)})^{-1} = ((P + k + c^{(1)})^d)^s \cdot (P + k + c^{(1)})^{2^t} = \alpha^s \cdot (P + k + c^{(1)})^{2^t}$, because the first term is constant and the second one is linear. We then obtain γ linearly independent equations about the variables $\{k_i\}_{i \in [n]}$ from the equation $(P + k + c^{(1)})^{2^{2w}} = (P + k + c^{(1)})^{(2^w + 1)(2^w - 1)} \cdot (P + k + c^{(1)}) = \alpha^{2^w - 1} \cdot (P + k + c^{(1)})$. Based on experimental evidence, the value of γ is always $n - 2w$. Consequently, each variable $\{k_i\}_{i \in [n]}$ can be expressed as a linear combination of $n - \gamma = 2w$ free variables $v \triangleq \{v_i\}_{i \in [2w]}$. The degree of freedom in this case is $|v| = 2w$.

Deriving Quadratic Equations from the Second S-Box. We can express the input of the second S-box as a linear combination of v . Similarly, the output of the second S-box, which can be obtained by computing $M_2^{-1}(k + C)$ backward, can also be expressed as a linear combination of v . For convenience, we denote the input and the output of the second S-box as $L_1(v)$ and $L_2(v)$, respectively. It is well-known that one can find overdefined quadratic equations from the multiplicative inverse function over \mathbb{F}_{2^n} [12]. Thus we can then derive $3n$ quadratic equations:

$$L_1(v) \cdot L_2(v) = 1, \quad L_1^2(v) \cdot L_2(v) = L_1(v), \quad L_1(v) \cdot L_2^2(v) = L_2(v). \quad (4)$$

After substituting the variables $\{k_i\}_{i \in [n]}$ with $|v|$ new free variables v_i , we obtain $3n$ quadratic equations. Let m denote the number of linearly independent quadratic equations. If we choose the parameters appropriately such that $m = |v| + |v|(|v| - 1)/2$, we can solve these equations using the linearization technique.

The Cost of Expressing Quadratic Equations by Free Variables. We first discuss the number of terms in the multiplication of two polynomials modulo an irreducible polynomial.

Let $a(x) = \sum_{i \in [n]} a_i x^i$ and $b(x) = \sum_{i \in [n]} b_i x^i$ be two polynomials, where the coefficients a_i, b_i are defined over some field \mathbb{F} , and x is called the indeterminate. If we consider the polynomials as abstract entities only, which are never evaluated, then the product $a(x) \cdot b(x) = \sum_{i \in [n], j \in [n]} a_i b_j x^{i+j}$ has exactly n^2 terms. However, when applying modular reduction operations by an irreducible polynomial, the number of terms in the product may increase, which makes it difficult to provide a formula to calculate the exact number of terms. In RAIN, the irreducible polynomials for the three security parameters are fixed (they are exactly the same for AIM), so we can compute the exact terms of the multiplication modulo the corresponding irreducible polynomials. Our computed results for the number of terms in the modulo multiplication over the three finite fields \mathbb{F}_{2^n} , $n \in \{128, 192, 256\}$, are presented in Table 2. Based on our findings, we are able to safely assume that the number of terms after applying modulo operations will not exceed $3n^2$.

Table 2. Number of terms in multiplication modulo irreducible polynomials for RAIN and AIM.

Finite Field	n	n^2	#Terms
$\mathbb{F}_{2^{128}}$	128	16,384	40,832
$\mathbb{F}_{2^{192}}$	192	36,864	91,936
$\mathbb{F}_{2^{256}}$	256	65,536	163,580

Let us backtrack to the phase of deriving quadratic equations for the second S-box. We observe that we can rewrite the quadratic equations in Eq. (4) uniformly as

$$\mathcal{L}_1(v) \cdot \mathcal{L}_2(v) = \mathcal{L}_3(v)$$

where $\mathcal{L}_i(v)$ ($i = 1, 2, 3$) is a set of linear functions of free variables v . This means that the maximum number of terms that we will need to represent using new free variables is $3n^2|v|^2 + n|v|$. In the following complexity estimation, the cost of this step will be estimated as $\mathcal{O}(n^2|v|^2)$. However, we will provide a discussion on the impact of the hidden constant in the big \mathcal{O} notation on the complexity of our attack.

Complexity Analysis. There are $(|\mathbb{F}_{2^n}| - 1)/d$ possible choices for the guess of α . For each guess, it takes $\mathcal{O}(n^3)$ time to find the free variables. Then it

costs $\mathcal{O}(n^2|v|^2)$ time to express the $3n$ quadratic equations in terms of the free variables. Finally, we find and solve m linearly independent quadratic equations in $\mathcal{O}(m^3)$ time. The total time complexity is counted as $T_{\text{bit}} = (2^n - 1)/d \cdot \max\{n^3, n^2|v|^2, m^3\}$. To express the complexity in terms of the number of encryptions, we divide by n^3 to obtain $T = T_{\text{bit}}/n^3$. Since our approach only requires storing the equations, the memory complexity can be omitted.

Parameter Sets of Our Attacks. Our detailed results on all three versions of two-round RAIN are listed in Table 3. We note that parameter s is involved in the equation $2^n - 2 = d \cdot s + 2^t$, thus can easily be obtained by checking $(2^n - 2 - 2^t) \bmod d = 0$. Due to its excessive length and it is not directly involved in the formula for the computation of complexity, we omit s from Table 3.

Table 3. Results for two-round RAIN. n denotes both the block size and the security parameter, r is the number of algorithmic rounds, $d = 2^w + 1$ represents a divisor of $2^n - 1$, t is defined by $d \cdot s + 2^t = 2^n - 2$, $|v|$ denotes the number of free variables, m denotes the number of variables in linearized quadratic equations, and T is the recalculated time complexity of our attack.

Scheme	n	r	d	t	w	$ v $	m	T
RAIN-128	128	2	$2^8 + 1$	8	8	16	136	$2^{120.3}$
RAIN-192	192	2	$2^{16} + 1$	16	16	32	528	$2^{180.4}$
RAIN-256	256	2	$2^{16} + 1$	16	16	32	528	$2^{243.1}$

The Hidden Constant Factor in Big \mathcal{O} and the Impacts. In the estimation of the time complexity of our attack, we omit the constant factor behind the big \mathcal{O} notations and consider the most dominant part only. We argue that such estimation is proper both in theory and in practice.

In the step of expressing the quadratic equations with new free variables, the maximum number of terms that we will need to represent is $3n^2|v|^2 + n|v|$. Therefore, to express three sets of equations as free variables in the RAIN, the cost will be

$$3 \times (3n^2|v|^2 + n|v|) = 9n^2|v|^2 + 3n|v| \leq 10n^2|v|^2. \quad (5)$$

The steps of deriving free variables and solving quadratic equations are basically the execution of Gaussian elimination. We estimate the concrete bit complexity as n^3 and m^3 , respectively. However, we point out they can be solved in $\mathcal{O}(n^\omega)$ and $\mathcal{O}(m^\omega)$ time respectively by using more sophisticated methods, such as the Strassen algorithm [40] ($\omega \approx 2.807$) or Coppersmith-Winograd algorithm [9] ($\omega \approx 2.376$). In recent algebraic attacks, both $\omega = 2.37$ and 2.8 are used [1, 31].

On the one hand, the constant in Eq. (5) is just an upper bound which is quite loose. Besides, the constants are affected by many factors, such as the algorithm, the implementation, and the computing architecture. So it is meaningful to estimate the time complexity by ignoring constants and focusing on the most dominant part.

Toy Example. We provide a concrete example for the attack on two-round RAIN, which can be found in Appendix A.2.

Negligible Failure Probability. It is worth pointing out that the equation $y = x^{|\mathbb{F}|-2}$ can be represented as $xy = 1$ only when $x \neq 0$. Fortunately, this probability is exponentially small. In particular, in the context of an MPC-in-the-head-based signature scheme such as [6, 14, 15, 20, 26], the input of the S-box must be non-zero, or else it would be rejected and re-sample again. Therefore, this issue is minor and can be safely ignored.

On Three- or More-Round RAIN. At present, our attack strategy is not effective for three or more rounds of RAIN. Following the strategy of our attack for reduced rounds of RAIN, we linearize the S-box in the first round as before. In order to express the relation between the S-boxes in the second and the last round, the introduction of new variables is required. As a result, the resulting equations become too complex to be solved efficiently because the number of unknown variables increases significantly.

4.3 Description of AIM

Let x, y be the input and output respectively. Let ℓ be the number of S-boxes of the first layer. Let $e_1, \dots, e_\ell, e_* \in \mathbb{N}$ be some constant. The S-boxes of AIM are exponentiation by Mersenne numbers, denoted as $\text{Mer}[e] : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$. More precisely, it is

$$\text{Mer}[e](x) = x^{2^e - 1}.$$

As an extension, $\text{Mer}[e_1, \dots, e_\ell] : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}^\ell$ is defined by

$$\text{Mer}[e_1, \dots, e_\ell](x) = \text{Mer}[e_1](x) || \dots || \text{Mer}[e_\ell](x).$$

The affine layer, denoted as Lin , is multiplication by an $n \times \ell n$ random binary matrix A_{iv} followed by an addition of a random constant $b_{\text{iv}} \in \mathbb{F}_{2^n}$. Both A_{iv} and b_{iv} are generated by an extendable output function (XOF) with a (public) initial vector iv .

Overall, the $\text{AIM}_{\text{iv}} : \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ function is defined by

$$\text{AIM}_{\text{iv}}(x) = \text{Mer}[e_*] \circ \text{Lin}[\text{iv}] \circ \text{Mer}[e_1, \dots, e_\ell](x) \oplus x.$$

A graphical overview of the construction is shown in Fig. 4.

Parameter Sets. The recommended sets of parameters of AIM are listed in Table 4. As a one-way function used in signature schemes, y and iv are public, x is private. The irreducible polynomials for extension fields $\mathbb{F}_{2^{128}}$, $\mathbb{F}_{2^{192}}$ and $\mathbb{F}_{2^{256}}$ are the same as those used in RAIN.

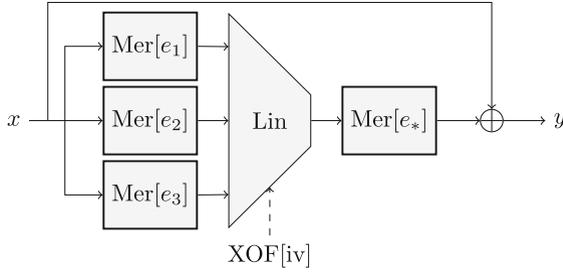


Fig. 4. The AIM-based one-way function with $\ell = 3$.

Table 4. Recommended sets of parameters of AIM.

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

4.4 Attack on AIM-III

We discovered a flaw in the design of AIM: the inputs of the S-boxes in the first nonlinear layer are identical, with no offset. This makes it possible for an attacker to linearize all the S-boxes in the first nonlinear layer at the same time. For AIM-III (the 192-bit version), we are able to exploit this flaw to mount a non-trivial attack.

In AIM-III, the first nonlinear layer comprises two S-boxes, namely $x^{2^{e_1}-1}$ and $x^{2^{e_2}-1}$, where $e_1 = 5$ and $e_2 = 29$. By guessing the value of $x^d = \alpha$ for $d = 45$, we can linearize these S-boxes using $x^{2^{e_i}-1} = (x^d)^{s_i} \cdot x^{2^{t_i}}$, $i \in \{1, 2\}$. (Note that $x^{2^{e_1}-1} = x^{2^{e_1}-1+|\mathbb{F}_{2^n}|-1}$)

The subsequent steps of the attack are similar to those used for the two-round RAIN. We can obtain $\gamma = 180$ linearly independent equations involving variables $\{x_i\}_{i \in [n]}$ from the equations $x^{2^{12}} = (x^{45})^{91} \cdot x = \alpha^{91} \cdot x$. Thus, we can express each $\{x_i\}_{i \in [n]}$ as a linear combination of $n - \gamma = 12$ free variables, denoted as $v \triangleq \{v_i\}_{i \in [12]}$. Consequently, the degree of freedom is $|v| = 12$.

The input of the second non-linear layer can be expressed as a linear combination of the free variables v . Similarly, the output of the second non-linear layer, obtained by computing $x + y$ in the backward pass, can also be expressed as a linear combination of v . Let us denote the input and output of the second non-linear layer as $L_1(v)$ and $L_2(v)$ respectively. Then, we can write n quadratic equations:

$$L_1(v) \cdot L_2(v) = L_1(v)^{2^{e^*}}$$

To solve the system of n quadratic equations, we can substitute the variables $\{x_i\}$ with $|v|$ new free variables $\{v_i\}$. This results in n quadratic equations in

the variables v_i . Let the number of linearly independent quadratic equations be denoted by m . We have found that $m = |v| + |v|(|v| - 1)/2$. Hence, we can use linearization techniques to solve these equations.

Complexity Analysis. The analysis is almost identical to the one for the attack on two-round RAIN. The only difference for AIM resides in the cost of expressing the quadratic equations arising from the second S-box by free variables. Following the analysis method for RAIN in Sect. 4, we conclude that the new cost of expressing the quadratic equations is $3n^2|v|^2 + n|v| \leq 4n^2|v|^2$.

Parameter Sets of Our Attack. Our results on AIM are listed in Table 5. We omit the parameter s_i due to its excessive length. We note that s_i is related to the equation $2^{e_i} - 1 = d \cdot s_i + 2^{t_i}$, where $(2^{e_i} - 1 - 2^{t_i}) \bmod d = 0$, can be easily verified.

Table 5. Results for AIM. n denotes both the block size and the security parameter, $d = 2^w + 1$ represents a divisor of $2^n - 1$, t_i is defined by $d \cdot s_i + 2^{t_i} = 2^{e_i} - 1$ for $i = 1, 2$, $|v|$ denotes the number of free variables, m denotes the number of variables in linearized quadratic equations, and T is the recalculated time complexity of our attack.

Scheme	n	d	t_1	t_2	$ v $	m	T
AIM-I	128	5	1	1	4	10	$2^{125.7}$
AIM-III	192	45	8	8	12	78	$2^{186.5}$
AIM-V	256	3	0	0	2	3	$2^{254.4}$

On AIM-I and AIM-V. We also analyzed the security of AIM-I and AIM-V using a similar technique for RAIN. We found that only $d = 3$ or $d = 5$ is a suitable choice of the divisor of $2^n - 1$. By substituting these parameters into the complexity formula, we obtain complexity estimations for AIM-I and AIM-V as $2^{125.7}$ and $2^{254.4}$ encryptions respectively. These complexities are only about $3 \sim 5$ times better than a brute-force attack, which is not considered as significant, so we do not take them as our contribution.

5 Implementation and Final Remarks

5.1 Experimental Verification of Our Attacks

We implement attacks on two-round RAIN and AIM-III described in Sect. 4. As mentioned earlier, our attacks require a sufficient number of linearly independent linear equations and quadratic equations. However, providing an explicit formula for determining the number of linearly independent equations from the equation

$x^d = \alpha$ is challenging. Therefore, we resort to programming to determine these equations. Our implementation can be found at

<https://github.com/kzoacn/LargeSbox/>.

In our code, we randomly sample an instance for either RAIN or AIM, which includes round constants and linear layers. We then randomly pick $\alpha \in \{x^d : x \in \mathbb{F}_{2^n}^*\}$ and determine the number of linearly independent linear equations in this instance. Next, we identify the free variables and substitute them into the final quadratic equations. Finally, we verify that the number of linearly independent quadratic equations is sufficient to solve all the unknown variables.

We also estimate the complexity concretely. Both of our approach and brute force follow the Guess-and-Determine framework, which allowed us to estimate the complexity by multiplying the duration of each trial by the total number of trials. The experimental outcomes align well with the theoretical predictions. However, it is worth pointing out that creating fair comparisons is challenging due to the impact of engineering optimizations on concrete complexity.⁴

5.2 Restoring the Security of AIM

Preventing the Simultaneous Linearization of Multiple S-boxes. Our attack on full AIM-III depends on that we can linearize both the S-boxes in the first non-linear layer simultaneously. Based on this, for the AIM cipher, we can easily recommend implementing a patch to enhance its resistance against our attack. The proposed patch involves incorporating offsets for each S-box in the first non-linear layer, for instance, XORing a random constant c_i before the operation of the S-box $\text{Mer}[e_i]$, as illustrated in Fig. 5. The constant c_i 's might be generated by an XOF, which is of low cost. By applying this patch, our attack is effectively mitigated as we can no longer linearize the S-boxes in the first layer simultaneously. We are not aware of attacks arising from involving extra random constants before the S-boxes in the setting of a single plaintext/ciphertext pair. Moreover, this patch can be seamlessly integrated into the MPC-in-the-head paradigm with minimal additional cost.

Restricting n to be Odd. We are aware that our attack highly relies on a foundational fact that the power function in RAIN and AIM can be represented as $(x^d)^s \cdot x^{2^t}$, where d is a divisor of $2^n - 1$. Recall that for an even n , $2^n - 1$ has a special factorization $2^n - 1 = (2^{n/2} + 1) \cdot (2^{n/4} + 1) \cdots (2^{n/2^t} - 1)$, where t is the maximal value of i such that 2^i divides n . However, this factorization can not be applied to an odd n . Therefore, the easiest way to prevent our attack is simply restricting n to an odd number.

⁴ We refer interested readers to a faster implementation utilizing SIMD instructions for RAIN (<https://github.com/IAIK/rainier-signatures>).

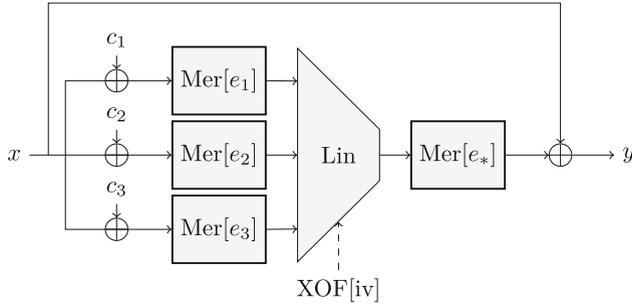


Fig. 5. Fixed AIM one way function with $\ell = 3$.

Increasing the Number of Rounds. A third alternative is to increase the number of rounds used in AIM, which can be achieved by appending more S-box operations after $\text{Mer}[e_*]$, such that the solving complexity of the system of equations is high enough to make the scheme secure against our attack. However, the main disadvantage of this approach is the loss of efficiency because applying more S-boxes would result in a higher number of multiplications, which might be detrimental for the signature schemes that are built upon AIM.

5.3 RAIN

As has been emphasized earlier, our attacks do not affect the security of the signature scheme RAINIER which uses three or more rounds of RAIN. However, we point out that for an even n , the essential decomposition of the multiplicative inverse function always holds. With a divisor d of $2^n - 1$, $\{x^d : x \in \mathbb{F}_{2^n} \setminus \{0\}\}$ is a subgroup of the multiplicative group $\mathbb{F}_{2^n}^*$ of the finite field \mathbb{F}_{2^n} . Therefore x^d has only $|\mathbb{F}_{2^n}^*|/d$ possible choices. Linearization of one round S-box can easily be accomplished. This might potentially be combined with other techniques, to threaten the security of RAIN.

5.4 On Other Relevant Symmetric Primitives

RAIN shares the similarity to MiMC [2], JARVIS [4] and instances of *Vision* [3], using a single large S-box covering the entire permutation state. While we have not discovered more novel attacks on the aforementioned ciphers based on the observation in this paper, we anticipate that it will contribute valuable insights to the understanding of these symmetric primitives.

6 Conclusion

In the past years, there have been remarkable advances in MPC-friendly symmetric primitive-based signatures. Constructing non-linear layers (S-boxes) by power mappings has proven advantageous in such signature schemes, exemplified

by the efficiency of RAINIER and AIMer compared to signature schemes based on other symmetric primitive-based ciphers. However, since our attacks on RAIN and AIM only exploit properties of power mappings over \mathbb{F}_{2^n} , which are actually independent of the choice of linear layers, one should be careful with using power mappings as the only non-linear components. We stress that we do not mean to suggest that power mappings should be avoided as a base structure for symmetric primitives, since several of the proposed schemes have useful properties in relevant use cases, in particular over \mathbb{F}_p where p is a very large prime. Rather, we emphasize that more thorough cryptanalysis is needed to ensure that the proposed primitives are secure, and hope to see more work in this direction.

Acknowledgments. Yu Yu was supported by the National Key Research and Development Program of China (Grant No. 2020YFA0309705) and the National Natural Science Foundation of China (Grant Nos. 62125204 and 92270201). This work was also supported in part by the National Key Research and Development Program of China (Grant No. 2018YFA0704701) and the Major Program of Guangdong Basic and Applied Research (Grant No. 2019B030302008). This work has been supported by the New Cornerstone Science Foundation through the XPLOER PRIZE.

A Toy Examples of Attacks on Reduced-Round RAIN

In this section, we concretely present toy examples for one-round RAIN and two-round RAIN, which would help the readers understand our attacks via examples. We choose the field \mathbb{F}_{2^4} with irreducible polynomial $f(x) = x^4 + x + 1$.

Notation. For an element $a_0 + a_1x + a_2x^2 + a_3x^3$ of field \mathbb{F}_{2^4} , $a_i \in \mathbb{F}_2$, we may express it in several equivalent forms:

1. Polynomial representation, i.e. $a_0 + a_1x + a_2x^2 + a_3x^3$.
2. Binary representation, i.e. $a_0a_1a_2a_3$.
3. Vector of \mathbb{F}_2 , i.e. (a_0, a_1, a_2, a_3)

Concretely, $1 + x + x^3$ is equivalent to 1101 or $(1, 1, 0, 1)$. And 1 is equivalent to 1000 or $(1, 0, 0, 0)$. We also use the notation $(1, 1, 0, 1)^2 = (1 + x + x^3)^2 = 1 + x^3 = (1, 0, 0, 1)$

Power Function. We first show the square function over \mathbb{F}_{2^4} is linear.

$$\begin{aligned} & (a_0 + a_1x + a_2x^2 + a_3x^3)^2 \bmod f(x) \\ &= (a_0 + a_1x^2 + a_2x^4 + a_3x^6) \bmod f(x) \\ &= a_0 + a_1x^2 + a_2(x + 1) + a_3(x^3 + x^2) \\ &= (a_0 + a_2) + a_2x + (a_1 + a_3)x^2 + a_3x^3 \end{aligned}$$

So we can use the notation $(a_0, a_1, a_2, a_3)^2 = (a_0 + a_2, a_2, a_1 + a_3, a_3)$. Similarly, we can have $(a_0, a_1, a_2, a_3)^4 = (a_0 + a_1 + a_2 + a_3, a_1 + a_3, a_2 + a_3, a_3)$.

Parameter Sets. We choose $P = 0000, c^{(1)} = 0010, c^{(2)} = 0001, k = 0100$ and

$$M_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad M_2^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

For convenience, we precompute some values and list them on Table 6. Let $d = 3$. As you can see, $x^d, x \neq 0$ takes only $15/3 = 5$ possible choices.

Table 6. Table for \mathbb{F}_{2^4} .

x	x^{-1}	$M_1(x)$	$M_2(x)$	x^3
0000	0000	0000	0000	0000
1000	1000	1001	1001	1000
0100	1001	0100	1010	0001
1100	0111	1101	0011	1111
0010	1011	1010	0100	0011
1010	1101	0011	1101	0101
0110	1110	1110	1110	1000
1110	0110	0111	0111	1000
0001	1111	0001	0001	0101
1001	0100	1000	1000	1111
0101	0011	0101	1011	1111
1101	1010	1100	0010	0011
0011	0101	1011	0101	0001
1011	0010	0010	1100	0101
0111	1100	1111	1111	0001
1111	0001	0110	0110	0011

A.1 One-Round RAIN

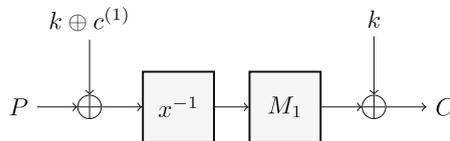


Fig. 6. RAIN permutation with $r = 1$.

The encryption of one-round RAIN (Fig. 6) is :

$$P = 0000 \xrightarrow{\oplus k \oplus c^{(1)}} 0110 \xrightarrow{x^{-1}} 1110 \xrightarrow{M_1(\cdot)} 0111 \xrightarrow{\oplus k} 0011 = C. \tag{6}$$

We guess the value of $(P + k + c^{(1)})^3$ over $(2^4 - 1)/3 = 5$ possible choices. Suppose we guess $(P + k + c^{(1)})^3 = 1$. The following steps will be repeated for each guess, here we only show the execution for the correct guess which is $(P + k + c^{(1)})^3 = 1$. So we have

$$\begin{aligned} (P + k + c^{(1)})^{-1} &= (P + k + c^{(1)})^{14} \\ &= (P + k + c^{(1)})^{3 \cdot 4} \cdot (P + k + c^{(1)})^2 \\ &= (P + k + c^{(1)})^2 \\ &= (k_0, k_1, k_2 + 1, k_3)^2 \\ &= (k_0 + k_2 + 1, k_2 + 1, k_1 + k_3, k_3) \end{aligned} \tag{7}$$

$$\begin{aligned} \xrightarrow{M_1(\cdot)} & (k_0 + k_1 + k_2 + k_3 + 1, k_2 + 1, k_1 + k_3, k_0 + k_2 + k_3 + 1) \\ \xrightarrow{\oplus k} & (k_1 + k_2 + k_3 + 1, k_1 + k_2 + 1, k_1 + k_2 + k_3, k_0 + k_2 + 1) \\ &= (0, 0, 1, 1) \\ &= C \end{aligned}$$

By solving the linear equations, we can obtain two candidate keys $k^* = 0100$ and $k^{**} = 0010$. Given the public input P and output C , it is easy to check by executing the one-round encryption in Eq. (6) that k^* is the correct key.

A.2 Two-Round RAIN

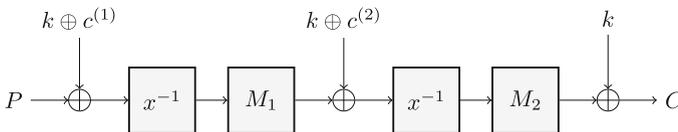


Fig. 7. RAIN permutation with $r = 2$.

We follow the notations and the parameter choices as in the attack on one-round RAIN. The encryption of two-round RAIN (Fig. 7) is :

$$\begin{aligned} P = 0000 & \xrightarrow{\oplus k \oplus c^{(1)}} 0110 \xrightarrow{x^{-1}} 1110 \xrightarrow{M_1(\cdot)} 0111 \\ & \xrightarrow{\oplus k \oplus c^{(2)}} 0010 \xrightarrow{x^{-1}} 1011 \xrightarrow{M_2(\cdot)} 1100 \xrightarrow{\oplus k} 1000 = C. \end{aligned}$$

- We linearize the first S-box by guessing $(P + k + c^{(1)})^3 = 1$ as in the previous attack on one-round. Then we have the following linear equations:

$$\begin{aligned}
 0 &= (P + k + c^{(1)})^4 + (P + k + c^{(1)}) \\
 &= (k_0, k_1, k_2 + 1, k_3)^4 + (k_0, k_1, k_2 + 1, k_3) \\
 &= (k_0 + k_1 + k_2 + k_3 + 1, k_1 + k_3, k_2 + k_3 + 1, k_3) + (k_0, k_1, k_2 + 1, k_3) \\
 &= (k_1 + k_2 + k_3 + 1, k_3, k_3, 0)
 \end{aligned}$$

This gives us two linearly independent equations:

$$\begin{cases} k_1 + k_2 = 1 \\ k_3 = 0 \end{cases}$$

Consequently, we can decide on two free variables k_0, k_2 and two basic variables $k_1 = k_2 + 1, k_3 = 0$.

- Next, we compute the input of the second S-box $M_1(P + k + c^{(1)})^{-1}$. Substituting M_1 as we choose and substituting the expression of $(P + k + c^{(1)})^{-1}$ as in Eq. (7), we get the following

$$\begin{aligned}
 M_1(P + k + c^{(1)})^{-1} &= (k_0 + k_1 + k_2 + k_3 + 1, k_2 + 1, k_1 + k_3, k_0 + k_2 + k_3 + 1) \\
 &\xrightarrow{\oplus k \oplus c^{(2)}} (k_1 + k_2 + k_3 + 1, k_1 + k_2 + 1, k_1 + k_2 + k_3, k_0 + k_2) \\
 &\xrightarrow{\text{using free vars.}} (0, 0, 1, k_0 + k_2)
 \end{aligned}$$

Backward, represent the output of the second S-box and we have

$$\begin{aligned}
 M_2^{-1}(k + C) &= (k_0 + k_2 + 1, k_2, k_1, k_0 + k_2 + k_3 + 1) \\
 &\xrightarrow{\text{using free vars.}} (k_0 + k_2 + 1, k_2, k_2 + 1, k_0 + k_2 + 1)
 \end{aligned}$$

Because the multiplication of the input and the output of the second S-box equals 1, thus we can write down the following quadratic equations:

$$\begin{aligned}
 (0, 0, 1, k_0 + k_2) \cdot (k_0 + k_2 + 1, k_2, k_2 + 1, k_0 + k_2 + 1) &= 1 \\
 \rightarrow (k_0 k_2 + 1, k_2, k_0 k_2 + k_0, k_2) &= 1
 \end{aligned}$$

By solving the linear equation, we can obtain a candidate key $k^* = 0100$. Given the public input P and the output C , it is easy to check that k^* is the correct key by executing two-round RAIN encryption.

References

1. Albrecht, M.R., et al.: Algebraic cryptanalysis of STARK-friendly designs: application to MARVELLOUS and MiMC. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 371–397. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_13

2. Albrecht, M., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: efficient encryption and cryptographic hashing with minimal multiplicative complexity. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 191–219. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_7
3. Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., Szepieniec, A.: Design of symmetric-key primitives for advanced cryptographic protocols. IACR Trans. Symm. Cryptol. 2020(3), 1–45 (2020). <https://doi.org/10.13154/tosc.v2020.i3.1-45>
4. Ashur, T., Dhooghe, S.: MARVELlous: a STARK-friendly family of cryptographic primitives. Cryptology ePrint Archive, Report 2018/1098 (2018). <https://eprint.iacr.org/2018/1098>
5. Banik, S., Barooti, K., Vaudenay, S., Yan, H.: New attacks on LowMC instances with a single plaintext/ciphertext pair. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 303–331. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_11
6. Baum, C., de Saint Guilhem, C.D., Kales, D., Orsini, E., Scholl, P., Zaverucha, G.: Banquet: short and fast signatures from AES. In: Garay, J.A. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 266–297. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-75245-3_11
7. Chase, M., et al.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, pp. 1825–1842. ACM Press (2017). <https://doi.org/10.1145/3133956.3133997>
8. Cheon, J.H., Lee, D.H.: Resistance of S-boxes against algebraic attacks. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 83–93. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25937-4_6
9. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, pp. 1–6 (1987)
10. Courtois, N., Debraize, B., Garrido, E.: On exact algebraic [non-]immunity of S-boxes based on power functions. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 06. LNCS, vol. 4058, pp. 76–86. Springer, Heidelberg (Jul 2006)
11. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_27
12. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of block ciphers with overdefined systems of equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_17
13. Daemen, J.: Limitations of the even-Mansour construction. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 495–498. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57332-1_46
14. de Saint Guilhem, C.D., De Meyer, L., Orsini, E., Smart, N.P.: BBQ: using AES in picnic signatures. In: Paterson, K.G., Stebila, D. (eds.) SAC 2019. LNCS, vol. 11959, pp. 669–692. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38471-5_27
15. de Saint Guilhem, C., Orsini, E., Tanguy, T.: Limbo: efficient zero-knowledge MPCitH-based arguments. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021, pp. 3022–3036. ACM Press (2021). <https://doi.org/10.1145/3460120.3484595>

16. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Key recovery attacks on 3-round Even-Mansour, 8-step LED-128, and full AES². In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 337–356. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_18
17. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Cryptanalysis of iterated even-Mansour schemes with two keys. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 439–457. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_23
18. Dinur, I., Liu, Y., Meier, W., Wang, Q.: Optimized interpolation attacks on LowMC. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 535–560. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48800-3_22
19. Dobraunig, C., Eichlseder, M., Mendel, F.: Higher-order cryptanalysis of LowMC. In: Kwon, S., Yun, A. (eds.) ICISC 2015. LNCS, vol. 9558, pp. 87–101. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30840-1_6
20. Dobraunig, C., Kales, D., Rechberger, C., Schafneggler, M., Zaverucha, G.: Shorter signatures based on tailor-made minimalist symmetric-key crypto. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022, pp. 843–857. ACM Press (2022). <https://doi.org/10.1145/3548606.3559353>
21. Dunkelman, O., Keller, N., Shamir, A.: Slidex attacks on the Even-Mansour encryption scheme. *J. Cryptol.* **28**(1), 1–28 (2015). <https://doi.org/10.1007/s00145-013-9164-7>
22. Gupta, K.C., Ray, I.G.: Finding biaffine and quadratic equations for s-boxes based on power mappings. *IEEE Trans. Inf. Theory* **61**(4), 2200–2209 (2015). <https://doi.org/10.1109/TIT.2014.2387052>
23. Hülsing, A., et al.: SPHINCS⁺. Technical report, National Institute of Standards and Technology (2022). <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
24. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Johnson, D.S., Feige, U. (eds.) 39th ACM STOC, pp. 21–30. ACM Press (2007). <https://doi.org/10.1145/1250790.1250794>
25. Katz, J., Kolesnikov, V., Wang, X.: Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018, pp. 525–537. ACM Press (2018). <https://doi.org/10.1145/3243734.3243805>
26. Kim, S., et al.: AIM: symmetric primitive for shorter signatures with stronger security. *Cryptology ePrint Archive*, Report 2022/1387 (2022). <https://eprint.iacr.org/2022/1387>
27. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 19–30. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_2
28. Liu, F., Isobe, T., Meier, W.: Cryptanalysis of full LowMC and LowMC-M with algebraic techniques. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12827, pp. 368–401. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84252-9_13
29. Liu, F., Mahzoun, M.: Algebraic attacks on RAIN and AIM using equivalent representations. *IACR Cryptol. ePrint Arch.* p. 1133 (2023). <https://eprint.iacr.org/2023/1133>
30. Liu, F., Meier, W., Sarkar, S., Isobe, T.: New low-memory algebraic attacks on LowMC in the Picnic setting. *IACR Trans. Symm. Cryptol.* **2022**(3), 102–122 (2022). <https://doi.org/10.46586/tosc.v2022.i3.102-122>

31. Liu, F., Sarkar, S., Meier, W., Isobe, T.: Algebraic attacks on rasta and dasta using low-degree equations. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 214–240. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_8
32. Liu, F., Sarkar, S., Wang, G., Meier, W., Isobe, T.: Algebraic meet-in-the-middle attack on LowMC. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part I. LNCS, vol. 13791, pp. 225–255. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-22963-3_8
33. Lyubashevsky, V., et al.: CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology (2022). <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
34. Mullen, G.L., Panario, D.: Handbook of Finite Fields. CRC Press, Boca Raton (2013)
35. Nawaz, Y., Gupta, K.C., Gong, G.: Algebraic immunity of s-boxes based on power mappings: analysis and construction. IEEE Trans. Inf. Theory **55**(9), 4263–4273 (2009). <https://doi.org/10.1109/TIT.2009.2025534>
36. Nikolić, Ivica, Wang, Lei, Wu, Shuang: Cryptanalysis of round-reduced LED. In: Moriai, Shiho (ed.) FSE 2013. LNCS, vol. 8424, pp. 112–129. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43933-3_7
37. Prest, T., et al.: FALCON. Technical report, National Institute of Standards and Technology (2022), available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
38. Rechberger, C., Soleimany, H., Tiessen, T.: Cryptanalysis of low-data instances of full LowMCv2. IACR Trans. Symm. Cryptol. **2018**(3), 163–181 (2018). <https://doi.org/10.13154/tosc.v2018.i3.163-181>
39. Srivastava, V., Baksi, A., Debnath, S.K.: An overview of hash based signatures. IACR Cryptol. ePrint Arch. p. 411 (2023). <https://eprint.iacr.org/2023/411>
40. Strassen, V., et al.: Gaussian elimination is not optimal. Numer. Math. **13**(4), 354–356 (1969)