Hindawi
IET Information Security
Volume 2024, Article ID 7457517, 11 pages
https://doi.org/10.1049/2024/7457517

Research Article

Differential Fault Attacks on Privacy Protocols Friendly Symmetric-Key Primitives: RAIN and HERA

Lin Jiao,[1] **Yongqiang Li**,[2,3] **Yonglin Hao**,[1] **and Xinxin Gong**[1]

[1]*State Key Laboratory of Cryptology, Beijing, China*
[2]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[3]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Lin Jiao; jiaolin_jl@126.com

Received 9 October 2023; Revised 5 March 2024; Accepted 18 March 2024; Published 27 March 2024

Academic Editor: Qichun Wang

As the practical applications of fully homomorphic encryption (FHE), secure multi-party computation (MPC) and zero-knowledge (ZK) proof continue to increase, so does the need to design and analyze new symmetric-key primitives that can adapt to these privacy-preserving protocols. These designs typically have low multiplicative complexity and depth with the parameter domain adapted to their application protocols, aiming to minimize the cost associated with the number of nonlinear operations or the multiplicative depth of their representation as circuits. In this paper, we propose two differential fault attacks against a one-way function RAIN used for Rainier (CCS 2022), a signature scheme based on the MPC-in-the-head approach and an FHE-friendly cipher HERA used for the RtF framework (Eurocrypt 2022), respectively. We show that our attacks can recover the keys for both ciphers by only injecting a fault into the internal state and requiring only one normal and one faulty ciphertext blocks. Thus, we can use only the practical complexity of $2^{26.6}/2^{28.8}/2^{30.4}$ bit operations to break the full-round RAIN with 128/192/256-bit keys. For full-round HERA with 80/128-bit key, our attack is practical with complexity the complexity of 2^{20} encryptions with about 2^{16} memory.

1. Introduction

With the rapid development of technologies, methods of data management, storage, and transformation have been significantly changed. Recently, some advanced protocols, such as multi-party computation (MPC), fully homomorphic encryption (FHE), and zero-knowledge (ZK) proof, have received a lot of attention in modern cryptography due to communication environments, such as big data and cloud computing. The development of novel symmetric-key primitives with MPC, FHE, and ZK applications has become a hot-spot research topic because of their importance for practical applications. Other than the traditional primitives, the primitives applicable to MPC, ZK, and FHE need to follow different efficiency metrics. In these applications, linear components can be considered "free," while nonlinear components can cause rapid noise growth and significantly increase execution time. Thus, nonlinear components are the most significant performance bottleneck in these applications. Specifically, MPC-friendly schemes aim to minimize the number of nonlinear operations required to evaluate these schemes in order to reduce the communication cost. ZK-friendly schemes aim to minimize the number of nonlinear operations required to prove these schemes. In contrast, FHE-friendly schemes aim to minimize the multiplication depth of their representation when the encryption and decryption process are represented as circuits, in order to improve the efficiency of homomorphic evaluations.

Some new symmetric schemes have been especially proposed for these privacy-preserving schemes. Some of these ciphers are designed based on operations over $\mathbb{F}_2$ usually with nonlinear functions of quadratic S-boxes or quadratic Boolean functions. For example, LowMC, proposed by Albretch et al. [1], is the first attempt to put forward a design that aims to minimize the number of AND gates and the AND depth. In 2016, the same group designed MiMC [2], a family of block ciphers, which directly operate with nonlinear function x^3 on the native finite field $\mathbb{F}_q$ the same with the protocol. This paved

a new way for such friendly ciphers, called Arithmetization-oriented ciphers, since many protocols naturally support operations in a large field $\mathbb{F}_q$, and converting operations over $\mathbb{F}_q$ into boolean operations is expensive. Multiple MPC-/ZK-/FHE-friendly ciphers were later introduced, such as FILP [3], Kreyvium [4], Jarvis [5], Rasta [6], Vision and Rescue [7], Poseidon [8], Ciminion [9], HERA [10], HADES [11], Reinforced Concrete [12], Rubato [13], Chaghri [14], Rain [15], Griffin [16], Anemoi [17], Pasta [18], and Hydra [19]. Especially, some ZK-friendly hash functions like Poseidon and Rescue have been adopted by the real-world application of blockchains due to the high efficiency, while some MPC-friendly primitives, such as LowMC, Rain, and the AIM [20] have been used to build some postquantum signature schemes, like Picnic and Banquet, with the MPC-in-the-head technique [21].

These specialized primitives pose new challenges in developing general methods or dedicated cryptanalysis techniques to understand their security. Meanwhile, insights on cryptanalysis of the new ciphers are constantly proposed to analyze the security of those ciphers, and many ciphers end up being found vulnerable to new attacks on their simple structure and less understood components. For example, there have been the guess and determine attack on FLIP [22], the Gröbner basis attack on Jarvis and Friday [23], various algebraic attacks on LowMC [24–29], the linearization attack on Rasta and Dasta [30], the high-order differential attacks on MiMC [31–33] and Chaghri [34, 35] over large finite fields, and the algebraic attacks on Rubato over rings [36]. There emerged some new cryptanalysis techniques to evaluate the security of these new primitives, and developed several useful cryptanalysis tools, among which the algebraic attacks usually have the most effective results.

In this paper, we focus on two such novel ciphers: RAIN designed for the postquantum signature scheme Rainier with the MPC-in-the-head technique (CCS 2022) [15] and the HE-friendly cipher HERA designed for RtF (Real-to-Finite-field) framework (ASIACRYPT 2021) [10]. The signature scheme of Rainier takes the public key as a single plaintext–ciphertext pair, and the private key as the secret key for the encryption of this plaintext–ciphertext pair. That is, the security of the signature scheme Rainier depends on the impossibility of recovering the secret key of RAIN by only one single known plaintext–ciphertext pair. In addition, the nonlinear components in RAIN are operated and implemented over large finite fields with high algebraic degree. Thus, to improve the efficiency of the signature scheme, the designers are very aggressive in the number of rounds chosen for the security of the RAIN cipher. The designers claimed that 3-round RAIN is secure and recommended the use of 4-round to further improve the security margin. Other than even 2-round RAIN cannot be cracked according to the designers' analysis, there are two algebraic attacks that have broken two rounds of RAIN recently [37, 38]. The RtF framework supports the CKKS scheme, which provides approximate arithmetic over real and complex numbers by combining the CKKS and FV homomorphic encryption schemes via stream ciphers with modular operations. HERA is an instantiation of such a stream cipher, which uses a simple randomized key schedule. HERA is an instantiation of such a stream cipher, which uses a simple randomized key schedule. Since HERA requires fewer random bits than the HE-friendly ciphers that use random linear layers, it outperforms in both client and server side. At present, the analysis of HERA is rare. Currently, there is an algebraic attack on HERA using multiple collisions [39].

Until now, very few studies related to side-channel analysis are conducted on such advanced protocol-friendly symmetric-key primitives, while through which these ciphers may be cracked efficiently, thus threatening the security of the system using these ciphers. Differential fault attack (DFA) is a widely employed semi-intrusive side-channel analysis, demonstrated by Biham and Shamir [40] on the symmetric-key ciphers in 1997 for DES. Since then, DFAs against symmetric-key ciphers have been extensively studied. DFA uses the differential information generated by introducing artificial faults when the cipher is running on the device to carry out the attack, which leverages computational errors to extract keys. In the traditional differential attacks, the attackers can only introduce differences to the public parameters of the cipher, such as the plaintext or the initialization vector. Instead, DFA is a more powerful attack model, where the attacker can also inject faults into the internal state of the cipher at some time instant in the encryption phase of the cipher. In order to apply DFAs on real devices, fault injection methods are required, including laser FI (laser-FI) [41], electromagnetic wave FI (EM-FI) [42], row-hammer attack (RHA) [43], voltage/clock glitches [44, 45], and others. Since then, there comes many different FI techniques for practical attacks [46–48]. Especially, high-level techniques, such as accurate memory address information or decapsulation, are required for laser-FI and RHA, and instead, the EM-FI is a practical technique for injecting faults while scanning the surface of the device, which directly helps the realization of out attacks.

For the DFA models, distinguishable differences in the generated ciphertext or keystream impacted by the faults introduced in the encryption phase of the cipher can be noticed by the attackers. The specific steps of DFAs for the attackers are first collect the desired number of normal ciphertext or keystream bits corresponding to an unknown key; then injects faults at random locations (since the frequently used techniques for FI cannot realize a very precise location) of the internal state in some fixed time instant and collects the required number of ciphertext or keystream bits affected by the faults; after that, perform statistical tests or exhaustively search on the keystream bits to determine the location of the injected fault, for the location of the injected fault is unknown. The statistical tests refer to [49–51]. In our work, we do not employ any statistical technique to determine the location of injected faults, since the required statistics appear to be random. Instead, we guess the location and value of the injected fault and perform the DFA.

In the context of privacy protocols, there are only two reference papers for the DFA attack on the friendly symmetric-key primitives. The first paper, published in 2021 [52], analyzes the DFA resilience of two stream ciphers: Kreyvium and FLIP. Both of these ciphers are suitable for use in FHE

FIGURE 1: The r-round RAIN.

schemes. The authors demonstrate that by injecting faults into the internal state of these ciphers, the secret key can be recovered. Specifically, for Kreyvium, they show that injecting 3-bit faults is sufficient to recover the key, while for FLIP, even a single-bit fault injection is enough. In the case of Kreyvium, they utilize statistical tests to pinpoint the location of the injected fault, whereas for FLIP, they rely on guessing the fault location. This attack is practical in terms of the time required to recover the key. The second paper, published in 2023 [53], extends the DFA analysis to two other FHE-friendly stream ciphers: Rasta and FiLIPDSM. Similarly, the authors demonstrate that by injecting a single-bit fault into the initial state of these ciphers, the secret key can be recovered. For one Rasta instance, which has a 219-bit block size, the attack requires only one block of normal and faulty keystream bits. For FiLIP-430, it requires 30,000 normal and faulty keystream bits to successfully recover the key. These papers highlight the importance of considering fault injection attacks when designing cryptographic primitives, especially those intended for use in privacy-preserving protocols. As this field of cryptography continues to evolve, it is crucial to remain vigilant against such threats and ensure that our cryptographic primitives are resilient against a wide range of attacks, including DFA.

1.1. Contribution. In this paper, we focus on two recently proposed MPC and FHE friendly ciphers: RAIN and HERA. We present a detailed security analysis of these ciphers with respect to DFA. We first identify specific vulnerabilities in the designs of RAIN and HERA that make them susceptible to DFA attacks. After that, we propose fault injection strategies tailored specifically for RAIN and HERA. Then, we develop practical key recovery attacks against RAIN and HERA, and the secret keys of these ciphers can be efficiently recovered. Finally, we present a comprehensive evaluation of the proposed attacks, discussing their impact on the security of RAIN and HERA. This evaluation includes a quantitative analysis of the required resources to successfully mount the attacks. The major contributions of this paper can be summarized as follows:

(i) In Section 3, we present a DFA on the RAIN cipher. Our analysis demonstrates that the secret key of RAIN can be efficiently recovered by introducing a single-bit fault into the internal state of the cipher. To mount this attack, we employ a generic DFA technique tailored for RAIN. We exhaustively explore various fault locations within the internal state of the cipher to identify the most effective points for fault injection. Our analysis reveals that, for the full-round concrete instances of RAIN with 128-bit, 192-bit, and 256-bit keys, the complexity of our attacks is practical. Specifically, using Gaussian elimination with a parameter $\omega = 2.8$ and a single known plaintext–ciphertext pair, we show that the key recovery complexity is approximately $2^{26.6}$ for the 128-bit version of RAIN, $2^{28.8}$ for the 192-bit version, and $2^{30.4}$ for the 256-bit version.

(ii) In Section 4, we introduce a DFA on the HERA cipher. Our analysis demonstrates that the secret key of HERA can be efficiently recovered by introducing a random word fault into the internal state of the cipher. Employing a generic DFA technique tailored for HERA, we exhaustively explore various fault values and word locations within the internal state. Our experiments reveal that, for the full-round concrete instances of HERA with 80-bit and 128-bit keys, our attacks are indeed practical with the key recovery complexity of approximately 2^{20} encryptions, requiring about 2^{16} memory and one keystream block.

1.2. Outline. In Section 2, we briefly describe the specifications of RAIN and HERA. Then, in Section 3 and Section 4, we present the DFA on RAIN and HERA, respectively. Finally, in Section 5, we conclude the paper by summarizing our findings.

2. Preliminaries

2.1. Design Specification of RAIN. RAIN is an MPC-friendly cipher used for the Signature scheme Rainer proposed at CCS 2022 [15].

The r-round RAIN is a keyed permutation $F_k(x)$ shown in Figure 1. The nonlinear operation S is the inverse function over $\mathbb{F}_{2^n}$, i.e.:

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

The round constants and linear layers are randomly generated according to some public parameters and fixed for each instance. Let c_i and $M_i \in (\mathbb{F}_2)^{n\times n}$ be the round constant added and the linear layer matrix acting on the internal state over $\mathbb{F}_2$ used in round i of RAIN, respectively. This matrix multiplication with such a binary matrix M_i can be transformed into a linearized polynomial $M_i \in \mathbb{F}_{2^n}[X]$, by a sum of n terms, each degree of which is a power of 2, i.e.:

$$M_i(X) = \sum_{j=0}^{n-1} a_{i,j} X^{2^j}, \tag{2}$$

for some known coefficients $a_{i,0}, \ldots, a_{i,n-1} \in \mathbb{F}_{2^n}$. In RAIN, it has been ensured $a_{i,j} \neq 0$ for each M_i, $j \in \{0, \ldots, n-1\}$, which means that the polynomial is of maximum degree and

Table 1: Concrete instances of RAIN.

Security	Field	Reduction polynomial
128-bit	$\mathbb{F}_{2^{128}}$	$X^{128}+X^7+X^2+X+1$
192-bit	$\mathbb{F}_{2^{192}}$	$X^{192}+X^7+X^2+X+1$
256-bit	$\mathbb{F}_{2^{256}}$	$X^{256}+X^{10}+X^5+X^2+1$

as dense as possible. Let s_i be the internal state in round i for $i \in \{0, \ldots, r\}$, then:

$$(s_0, s_r) : F_k(s_0) = s_r, \tag{3}$$

is the plaintext–ciphertext pair. Thus, we have the round function R_i defined as follows:

$$s_i = R_i(s_{i-1}) = \begin{cases} M_i \circ S(s_{i-1} \oplus k \oplus c_i), & \text{if } i < r \\ S(s_{i-1} \oplus k \oplus c_i) \oplus k, & \text{if } i = r \end{cases}. \tag{4}$$

That is, the input of each round is the XOR result of the output of the previous round with the round constant and the key. If it is not the last round, the output is a combination of linear layer operations and nonlinear operations; if it is the last round, there is only one more XOR operation with the key.

The concrete instances are defined in Table 1.

The security of RAIN is mainly based on the fact that the attacker only knows one plaintext–ciphertext pair under the same key. Therefore, the designer chose the number of rounds r to be 3 or 4 to ensure security under this restriction. Indeed, in the signature scheme Rainer, k is the secret key while the (s_0, s_r) is the public key. Therefore, the attack on RAIN is directly related to the security of Rainer.

According to the analysis of the designers of RAIN, even 2-round RAIN cannot be broken. However, attacks on 2-round RAIN one 128/192/256-bit key in the complexity of $2^{116}/2^{171}/2^{224}$ by equivalent representations are given in Liu et al.'s [38] study recently.

2.2. HERA. We denote $\mathbb{Z}_t$ with $\mathbb{Z} \cap (-t/2, t/2]$ for an integer t. The target stream cipher HERA for λ-bit security takes a secret key $k \in \mathbb{Z}_t^{16}$, a nonce $\mathrm{nc} \in \{0, 1\}^{\lambda}$, a counter $\mathrm{ctr} = 0, 1, \ldots, l-1$ as input and returns a keystream of needed l blocks $z \in (\mathbb{Z}_t^{16})^l$, where the nonce and counter are fed into an underlying extendable output function (XOF) that outputs an element in $(\mathbb{Z}_t^{16})^*$.

The process of HERA cipher is defined as follows:

$$\begin{aligned} \mathrm{HERA}[k, \mathrm{nc}\|\mathrm{ctr}](ic) = {} & \mathrm{Fin}[k, \mathrm{nc}\|\mathrm{ctr}, r] \circ \mathrm{RF}[k, \mathrm{nc}\|\mathrm{ctr}, 1] \\ & \circ \cdots \circ \mathrm{RF}[k, \mathrm{nc}\|\mathrm{ctr}, 1] \circ \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, 0](ic), \end{aligned} \tag{5}$$

where ic is a constant $(1, 2, \ldots, 16) \in \mathbb{Z}_t^{16}$, $\mathrm{RF}[k, \mathrm{nc}\|\mathrm{ctr}, i]$ is the ith round function and Fin is the final round function. Let the internal state be $s_i \in \mathbb{Z}_t^{16}$ for $i = 0, 1, \ldots, r$, which is also viewed as a 4×4-matrix over $\mathbb{Z}_t$. Then:

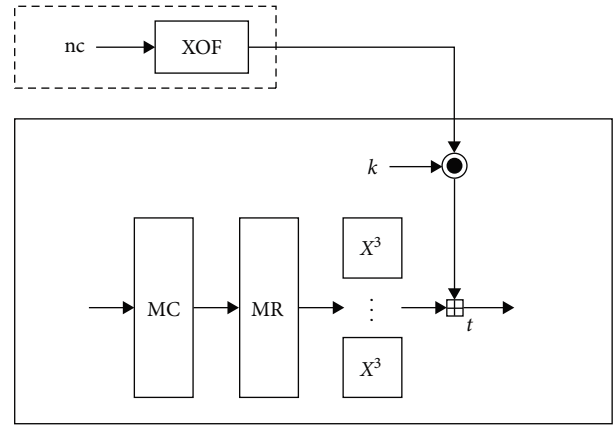

Figure 2: The round function of HERA.

$$\begin{aligned} s_0 &= \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, 0](ic), \\ s_i &= \mathrm{RF}[k, \mathrm{nc}\|\mathrm{ctr}, 1](s_{i-1}) \\ &= \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, i] \circ \mathrm{Cube} \circ \mathrm{MR} \circ \mathrm{MC}(s_{i-1}), \\ &\quad \text{for } i = 1, \ldots, r-1, \\ z_{\mathrm{ctr}} &= s_r = \mathrm{Fin}[k, \mathrm{nc}\|\mathrm{ctr}, r](s_{r-1}) \\ &= \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, r] \circ \mathrm{MR} \circ \mathrm{MC} \circ \mathrm{Cube} \circ \mathrm{MR} \circ \mathrm{MC}(s_{r-1}). \end{aligned} \tag{6}$$

The round function of HERA is shown in Figure 2.

Given a sequence $rc = (rc_0, \ldots, rc_r) \in (\mathbb{Z}_t^{16})^{r+1}$ of the outputs from XOF, which can be instantiated with a hash function like SHAKE256 [54], ARK is defined as follows:

$$\mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, i](x) = x + k \bullet rc_i, \tag{7}$$

for $i = 0, 1, \ldots, r$, and $x \in \mathbb{Z}_t^{16}$, where $\bullet$ (resp. $+$) denotes component-wise multiplication (resp. addition) modulo t.

Each linear layer is composed of MC and then MR, where MC (resp. MR) multiplies a certain 4×4 matrix:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}, \tag{8}$$

with entries in $\mathbb{Z}_t$ to each column (resp. row) of the internal state. The matrix results in MDS matrix over $\mathbb{Z}_t$ when the integer t is prime and larger than 17.

The nonlinear map Cube is defined as follows:

$$\mathrm{Cube}(x) = (x_0^3, \ldots, x_{15}^3), \tag{9}$$

for $x = (x_0, \ldots, x_{15}) \in \mathbb{Z}_t^{16}$. For the bijectivity, it is required that $\gcd(3, t-1) = 1$.

The designer presents two instances of HERA(r, t), HERA (4, 65537) for 80-bit security, and HERA (5, 65537) for 128-bit security, where r and t, respectively, denote the number of rounds and the modulus.

3. DFA on RAIN

This section describes our DFA on RAIN.

Since our target primitive RAIN is defined over a finite field F, there is an isomorphism between, so we focus only on Boolean equations.

In simple terms, DFA model is a fault analysis model based on specific assumptions, which is used to analyze the security of cryptographic algorithms. In this model, the attacker has the ability to restart cryptographic algorithms, inject transient faults, and observe the results of failures and normal operations. However, the attacker cannot precisely control where the fault occurs. Since the target primitive is defined over a specific finite field and there is an isomorphism between this field and another space, this allows us to transform the problem into the form of a Boolean equation for study. This transformation helps to simplify the analysis process and may reveal potential weaknesses of cryptographic algorithms in the face of fault attacks.

3.1. Underlying Assumptions for DFA. Our DFA model relies on the specific assumptions as follows:

(1) The attacker has the ability to restart the cipher with the same key and other public parameters as inputs.
(2) The attacker can inject a transient fault at a specific time instant during the course of the encryption/decryption of the cipher and monitor pairings of normal and fault ciphertexts.
(3) The attacker has tools needed to perform the fault injection of single-bit flip (e.g., laser-FI, RHA, etc.).
(4) The attacker cannot precisely specify the location of the fault injection, i.e., the fault occurs in a random position.

Our target primitive RAIN is defined over the finite field $\mathbb{F}_{2^n}$. Since there is an isomorphism between the field of $\mathbb{F}_{2^n}$ and the vector space of $\mathbb{F}_2^n$, we only focus on the Boolean equations. Moreover, since the internal state stored in memory is actually in the binary form, we inject a single-bit fault in the internal state before the last round. Since the fault injected position is unknown, we should exhaustive search the n differences e_i for $i = 1, \ldots, n$, which denote the n-dimensional unit vector.

3.2. Transformation from Univariate Polynomials over Finite Fields into Boolean Equations. For an object system of equations defined over a field of $\mathbb{F}_{2^n}$, we first convert them into a system of Boolean equations with certain known algebraic degree according to the isomorphism. This transformation allows us to solve them and then taking advantage of generic solving techniques already well-established for Boolean equations.

Let:

$$F(x) = \sum_{j=0}^{2^n-1} \mu_j x^j, \tag{10}$$

be a univariate polynomial in $\mathbb{F}_{2^n}[x]$, where $\mu_j \in \mathbb{F}_{2^n}$ is the coefficients. Given an irreducible polynomial $q(\alpha)$ with degree n in $\mathbb{F}_2[x]$, we have a basis of $\{1, \alpha, \ldots, \alpha^{n-1}\}$ for $\mathbb{F}_{2^n}$. Then:

$$x = \sum_{i=0}^{n-1} x_i \alpha^i,\ x_i \in \mathbb{F}_2, \text{ and } \mu_j = \sum_{i=0}^{n-1} u_{j,i} \alpha^i,\ u_{j,i} \in \mathbb{F}_2. \tag{11}$$

Let $j = \Sigma_{s=0}^{n-1} j_s 2^s$ in the binary form. Thus:

$$\begin{aligned} F(x) &= \sum_{j=0}^{2^n-1} \left(\sum_{i=0}^{n-1} u_{j,i} \alpha^i \right) \left(\sum_{i=0}^{n-1} x_i \alpha^i \right)^{\Sigma_{s=0}^{n-1} j_s 2^s} \mod q(\alpha) \\ &= \sum_{j=0}^{2^n-1} \left(\sum_{i=0}^{n-1} u_{j,i} \alpha^i \right) \prod_{s=0}^{n-1} \left(\sum_{i=0}^{n-1} x_i \alpha^{2^s} \right)^{j_s} \mod q(\alpha) \quad . \\ &= \sum_{j=0}^{n-1} f_j \alpha^j \end{aligned} \tag{12}$$

This polynomial F can be equivalently given by its n vectorial Boolean polynomials $f_1, \ldots, f_n$ with respect to n Boolean variables $x_1, \ldots, x_n$. Each Boolean polynomial f_j has an algebraic degree of $\max\{\mathrm{wt}(j) : u_j \neq 0\}$, where $\mathrm{wt}(j)$ denotes the number of ones in the binary representation of the integer j.

Thus, an exponential function of $x \to x^{2^i}$, for $i \geq 0$ over $\mathbb{F}_{2^n}$ corresponds to a linear vectorial Boolean function in terms of x for the isomorphism. In other words, this operation introduces no multiplication between Boolean variables.

3.3. DFA on RAIN. For mounting the DFA on RAIN, we inject a fault of single-bit flip in the internate state s_{r-1}, but the exact location of the fault injection is unknown. Therefore, all possible fault locations should be exhaustively tried. The attack process shown in Algorithm 1 requires only one block.

Given the guessed 1-bit fault location, known normal and faulty ciphertexts, we can set up a system of linear equations only in k. First, we consider whether $k = s_r$, and it can be trivially verified by if $F_{s_r} = s_r$. Thus, we always assume that $k \oplus s_r \neq 0$ in the following context. Thus, we can derive the following equation:

$$s_{r-1} \oplus k \oplus c_3 = \frac{1}{k \oplus s_r}. \tag{13}$$

Then according to the guessed difference fault, we have the following equation:

$$\Delta s_{r-1} = \frac{1}{k \oplus s_r} \oplus \frac{1}{k \oplus s_r'} = \frac{s_r \oplus s_r'}{k^2 \oplus (s_r \oplus s_r') \cdot k \oplus s_r \cdot s_r'}, \tag{14}$$

i.e.

$$k^2 \oplus (s_r \oplus s_r') \cdot k = (s_r \oplus s_r') \cdot \Delta s_{r-1}^{-1} \oplus s_r \cdot s_r'. \tag{15}$$

Since the exponential function of $x \to x^{2^i}$, for $i \geq 0$ over $\mathbb{F}_{2^n}$ corresponds to a linear vectorial Boolean function with

1: Collect the normal ciphertext s_r for an unknown key k on s_0.
2: Inject one-bit fault at a random position in the register of the state s_{r-1}
3: Collect the faulty ciphertext s'_r for the same key and public parameters
4: **for** $\Delta s_{r-1} = e_i, i = 1, \ldots, n$ **do**
5: Construct a system of Boolean equations involving the key k as unknown variables based on the normal ciphertext and the corresponding faulty ciphertexts
6: Solve this system of equations for solution $\tilde{k}$
7: **if** $F_{\tilde{k}}(s_0) = s_r$ **then**
8: **return** $k = \tilde{k}$;
9: **end if**
10: **end for**

Algorithm 1: DFA on RAIN.

respect to x, the above equation over $\mathbb{F}_{2^n}$ in k can be equivalently represented as n linear Boolean equations in k, named as follows:

$$L(x) = \tau, \tag{16}$$

where $\tau = ((s_r \oplus s'_r) \cdot \Delta s_{r-1}^{-1} \oplus s_r \cdot s'_r)$ is known.

Next, we can solve for the solution $\tilde{k}$ by Gaussian elimination, an algorithm used to solve systems of linear equations, with a complexity of n^ω. In the naive implementation, its time complexity is $\omega = 3$. In 1969, due to Strassen's divide and conquer algorithm by recursively factoring large matrices into smaller matrices and combining their results in a more efficient way [55], the upper bound on ω is updated to ω, and such an algorithm has a practical implementation in library [56]. Although there exists a more efficient algorithm [57], that further reduces the upper bound of to ω to less than 2.37, but this algorithm may not be useful in practice due to its large hidden constant factor.

As the location of the injected fault is unknown, we need to simulate this process for all n possible fault locations. Next, we need to filter out the secret key by confirming the following equation:

$$F_{\tilde{k}}(s_0) = s_r. \tag{17}$$

Since the time required to check the consistency of the keystream generated by the candidate keys is negligible, the DFA on full-round RAIN takes about $n^{\omega+1}$ bit operations in total.

Thus for the full-round concrete instances with 128/192/256-bit key, our attacks are practical with the complexity of $2^{26.6}/2^{28.8}/2^{30.4}$ with $\omega = 2.8$. We have tested our attacks and made the experimental verification.

We have performed our DFA attack simulations on RAIN with a 128-bit block size. We first collect the normal ciphertext s_r for an unknown key k on $s_0 = 0$. Next, we inject a single-bit fault at a random location in the state s_{r-1} by the tool such as laser-FI, RHA, etc. After that, we collect the faulty ciphertext s'_r for the same key and other public parameters. We assume the fault occurs in one bit of the internal state s_{r-1}. We then generate a system of equations involving the unknown key k from the normal and faulty ciphertexts as Equation (15). We can use NTL library/SageMath software to generate these equations as follows. To transform a linearized polynomial $L(X)$ in $\mathbb{F}_{2^n}[x]$ into a matrix M in $\mathbb{F}_2^{n\times n}$. We first compute the basis of the field $\mathbb{F}_{2^n}$ with ordered power $\{\beta_1, \beta_2, \ldots, \beta_n\} = \{\beta, \beta^2, \ldots, \beta^{n-1}\}$ and its dual basis $\beta'_1, \beta'_2, \ldots, \beta'_n$ such that $tr(\beta_i \beta'_j) = \delta_i, j$ for i in $[0, n]$, j in $[0, n]$. Here, $tr : \mathbb{F}_{2^n} \to \mathbb{F}_2$ denotes the trace function over $\mathbb{F}_{2^n}$ and δ_i, j denotes the Kronecker delta. Then, the element M_i, j at the jth column and the ith row of the matrix M is defined as follows:

$$M_i, j = tr\left(\beta'_i L\left(\beta_j\right)\right). \tag{18}$$

The cost of this transformation is approximately $n^2 + n^3$ operations of field. For example, we can construct the 128-bit matrix in only a few seconds on our ordinary computer. After generating these systems of equations, we use Gaussian elimination to find the solution of the key.

4. DFA on HERA

This section presents our DFA on HERA cipher.

4.1. Underlying Assumptions for DFA. Our DFA model relies on the first assumptions in Section 3.1 and the following assumptions:

(1) The attacker is able to change a single word to a random value with some EM-FI tools.
(2) The attacker cannot precisely specify the location of the fault injection, i.e., the fault occurs in a random word.

4.2. DFA on HERA. For mounting the DFA on HERA, we inject a fault in the internal state s_{r-1} and change a single word into a random unknown value, also the injected word location of the fault is unknown. Therefore, all 16 words of the internal state should be exhaustively tried. The attack process shown in Algorithm 2 requires only one block.

Given the guessed one-word fault location i and the value of the faulty word $s'_{r-1,i} \in \mathbb{Z}_t$, we have Δs_{r-1} with an only nonzero component as follows:

1: Collect the normal ciphertext s_r for an unknown key k on ic.
2: Inject only one word fault at a random position in the register of the state s_{r-1}
3: Collect the faulty ciphertext s'_r for the same key and same $nc\|ctr$
4: **for** each word position $i = 0, \ldots, 15$ **do**
5: **for** each value of the faulty word $s'_{r-1,i} \in \mathbb{Z}_t$ **do**
6: Construct 16 quadratic equations over $\mathbb{Z}_t$ involving the normal input x of the nonlinear map $Cube$ as unknown from the normal ciphertexts and the corresponding faulty ciphertexts
7: Solve the equations for $\tilde{x}$
8: Solve for the key $\tilde{k}$ according to $\tilde{k} = (MR \circ MC \circ Cube(\tilde{x}) + s_r) \bullet rc_r^{-1}$
9: **if** $\mathrm{HERA}[\tilde{k}, nc\|ctr](ic) = s_r$ **then**
10: **return** $k = \tilde{k}$;
11: **end if**
12: **end for**
13: **end for**

Algorithm 2: DFA on HERA.

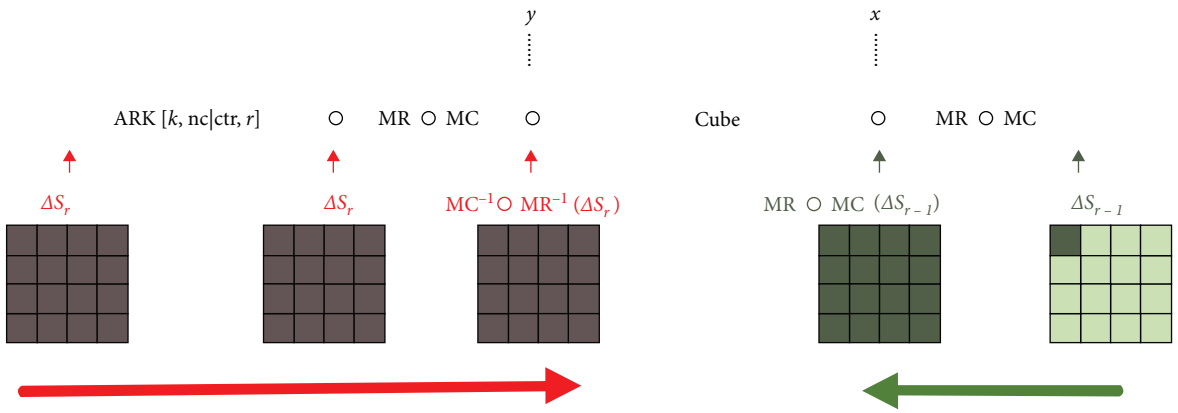

Figure 3: The propagation of the differential fault in the DFA on HERA.

$$\Delta s_{r-1,j} = \begin{cases} s'_{r-1,i} - s_{r-1,i}, & j = i \\ 0, & j = 0, \ldots, 15, \ j \neq i \end{cases}. \tag{19}$$

Let x, x' and y, y' be the normal and fault input, the normal and fault output of the nonlinear map Cube, respectively. Since MR and MC are linear operations, we have the following equation:

$$\Delta x = x' - x = \mathrm{MR} \circ \mathrm{MC}(\Delta s_{r-1}). \tag{20}$$

Similarly:

$$\begin{aligned} \Delta s_r = s'_r - s_r &= \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, r] \circ \mathrm{MR} \circ \mathrm{MC}(y') \\ &\quad - \mathrm{ARK}[k, \mathrm{nc}\|\mathrm{ctr}, r] \circ \mathrm{MR} \circ \mathrm{MC}(y) \\ &= \mathrm{MR} \circ \mathrm{MC}(y' - y). \end{aligned} \tag{21}$$

Thus:

$$\Delta y = y' - y = \mathrm{MC}^{-1} \circ \mathrm{MR}^{-1}(\Delta s_r). \tag{22}$$

The differential fault propagates as shown in Figure 3.

For each component $i = 0, \ldots, 15$ of the Cube map, we have the following equation:

$$\begin{aligned} \Delta y_i &= (x_i + \Delta x_i)^3 - x_i^3 \\ &= 3 \cdot \Delta x_i \cdot x_i^2 + 3 \cdot (\Delta x_i)^2 \cdot x_i + (\Delta x_i)^3, \end{aligned} \tag{23}$$

which is a quadratic equation over the finite field $\mathbb{Z}_t$ in the form of the following equation:

$$ax^2 + bx + c \equiv 0 \mod t. \tag{24}$$

Since the case of $t = 2$ is obvious, it is always assumed that the t is odd prime in the following. Let t does not divide a, i.e., $t \nmid a$. The solution of Equation (24) is the same as that of the following equation:

$$(2ax + b)^2 \equiv b^2 - 4ac \mod t, \tag{25}$$

since $t \nmid 4a$. Let $\gamma \equiv 2ax + b \mod t$, then Equation (25) is equivalent to the following equation:

$$\gamma^2 \equiv b^2 - 4ac \mod t. \tag{26}$$

That is Equations (25) and (26) both have solutions with the same solution number or no solutions. Therefore, we only need to discuss the equation in the form of the following equation:

$$\gamma^2 \equiv d \mod t. \tag{27}$$

When t divides d, i.e., $t|d$, Equation (27) only has a solution:

$$x \equiv 0 \mod t. \tag{28}$$

Thus, we assume $t\nmid d$.

Definition 1. Let prime $t>2$, d is an integer, $t\nmid d$. We call d a quadratic residue modulo t if the congruence Equation (27) has a solution, or a quadratic nonresidue modulo t if there is no solution for Equation (27).

For example, when $t=3$, $d=1 \mod 3$ is a quadratic residue of modulo 3 with solutions of ± 1 and $d=-1 \mod 3$ is a quadratic nonresidue of modulo 3. In general, the following conclusions are drawn.

Theorem 1. *In a reduced system of residues modular t, i.e.:*

$$\left\{-\frac{t-1}{2}, -\frac{t-1}{2}+1, \ldots, -1, 1, \ldots, \frac{t-1}{2}-1, \frac{t-1}{2}\right\}, \tag{29}$$

there are exactly $(t-1)/2$ *quadratic residues modulo t, and* $(t-1)/2$ *quadratic nonresidues modulo t. Moreover, if d is a quadratic residue modulo t, the number of solutions of the congruence Equation (27) is 2.*

Therefore, given some Δy_i and Δx_i, the probability that Equation (23) has solutions is as follows:

$$\Pr[\text{one solution}] + \Pr[\text{two solution}] = \frac{1}{t} + \frac{t-1}{2t} = \frac{t+1}{2t}. \tag{30}$$

Then, it takes a probability about $(\frac{t+1}{2t})^{16} \approx (1/2)^{16}$ to have solutions for the internal state x, by solving each Equation (23) for the component $i=0,\ldots,15$ of the Cube map, respectively. If there are solutions for x, the number of solutions for x is Equation (31) on average:

$$\sum_{i=0}^{16} \left(1 \cdot \frac{2}{t+1}\right)^i \left(2 \cdot \frac{t-1}{t+1}\right)^{16-i} \approx 2^{16}. \tag{31}$$

Given each solution $\tilde{x}$, we solve for the key $\tilde{k}$ according to the following equation:

$$\widetilde{k} = (\mathrm{MR} \circ \mathrm{MC} \circ \mathrm{Cube}(\widetilde{x}) + s_r) \bullet rc_r^{-1}. \tag{32}$$

Since the injected word location of the fault is unknown, and the fault changes a single word in the state s_{r-1} into a random value, there are Equation (33) candidates of $\tilde{k}$ should be exhaustively tried:

$$16 \cdot (t-1) \cdot \left(1 \cdot \frac{1}{t} + 2 \cdot \frac{t-1}{2t} + 0 \cdot \frac{t-1}{2t}\right)^{16} = 16(t-1). \tag{33}$$

We test if:

$$\mathrm{HERA}\left[\widetilde{k}, \mathrm{nc} \| \mathrm{ctr}\right](ic) = s_r, \tag{34}$$

and filter out the only k.

Next, we analyze the complexity. To solve Equation (23), we can build a table to store the quadratic residues d modulo t and their solutions γ according to the reduced system of residues modular t offline. Then, we only need to look up the table online according to the following equation:

$$d \equiv 9 \cdot (\Delta x_i)^4 - 12 \cdot \Delta x_i \cdot ((\Delta x_i)^3 - \Delta y_i) \mod t, \tag{35}$$

for

$$\widetilde{x} = \frac{\gamma - 3 \cdot (\Delta x_i)^2}{6 \cdot (\Delta x_i)}, \tag{36}$$

over $\mathbb{Z}_t$. It needs a number of $16 \cdot 16 \cdot (t-1)$ table look-ups, which is negligible compared with the entire encryptions. According to the above analysis, there are $16(t-1)$ candidates for $\tilde{x}$ and then for $\tilde{k}$ should be exhaustively tried by the entire encryption. Only one block of keystream can filter out the right key. Therefore for the DFA attack on HERA, the time complexity online is about $16(t-1)$ encryptions, the time complexity offline and the memory complexity to store the table is about t, and the data complexity is only one keystream block.

Thus for the full-round concrete instances with 80/128-bit key, our attacks are practical with the complexity of 2^{20} encryptions with about 2^{16} memory and one keystream block.

5. Conclusion

In this paper, we have proposed differential fault analysis on two recently designed symmetric-key ciphers for MPC and FHE, namely RAIN and HERA. We have shown that the key of both ciphers can be easily recovered in practical time just by injecting a single bit fault or a random word fault into the state register of the cipher according to a single plaintext–ciphertext pair. Through the study of these two friendly symmetric-key ciphers for advanced protocols that we analyzed, we strongly believe that this class of symmetric ciphers adapt to privacy-preserving protocols is vulnerable to differential fault analysis, and therefore, the design criteria and measures of resistance need to be paid attention to.

Data Availability

No underlying data were collected or produced in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Yongqiang Li and Xinxin Gong are supported by the National Natural Science Foundation of China (Grant Nos. 12371525 and 62202062), respectively.

References

[1] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner, "Ciphers for MPC and FHE," in *Advances in Cryptology-EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds., vol. 9056 of *Lecture Notes in Computer Science*, pp. 430–454, Springer, Sofia, Bulgaria, 2015.

[2] M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen, "MiMC: efficient encryption and cryptographic hashing with minimal multiplicative complexity," in *Advances in Cryptology-ASIACRYPT 2016*, vol. 10031 of *Lecture Notes in Computer Science*, pp. 191–219, Springer, Hanoi, Vietnam, 2016.

[3] P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet, "Towards stream ciphers for efficient FHE with low-noise ciphertexts," in *Advances in Cryptology-EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds., vol. 9665 of *Lecture Notes in Computer Science*, pp. 311–343, Springer, Vienna, Austria, 2016.

[4] A. Canteaut, S. Carpov, C. Fontaine et al., "Stream ciphers: a practical solution for efficient homomorphic-ciphertext compression," in *Fast Software Encryption*, vol. 9783 of *Lecture Notes in Computer Science*, pp. 313–333, Springer, Bochum, Germany, 2016.

[5] T. Ashur and S. Dhooghe, "Marvellous: a stark-friendly family of cryptographic primitives," *Cryptology ePrint Archive-IACR*, Article ID 1098, 2018.

[6] C. Dobraunig, M. Eichlseder, L. Grassi et al., "Rasta: a cipher with low anddepth and few ands per bit," in *Advances in Cryptology-CRYPTO 2018*, H. Shacham and A. Boldyreva, Eds., vol. 10991 of *Lecture Notes in Computer Science*, pp. 662–692, Springer, Santa Barbara, CA, USA, 2018.

[7] A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe, and A. Szepieniec, "Design of symmetric-key primitives for advanced cryptographic protocols," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 1–45, 2020.

[8] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger, "Poseidon: a new hash function for zero-knowledge proof systems," in *30th USENIX Security Symposium*, M. Bailey and R. Greenstadt, Eds., pp. 519–535, USENIX Association, 2021.

[9] C. Dobraunig, L. Grassi, A. Guinet, and D. Kuijsters, "Ciminion: symmetric encryption based on toffoli-gates over large finite fields," in *Advances in Cryptology-EUROCRYPT 2021*, A. Canteaut and F.-X. Standaert, Eds., vol. 12697 of *Lecture Notes in Computer Science*, pp. 3–34, Springer, Zagreb, Croatia, 2021.

[10] J. Cho, J. Ha, S. Kim et al., "Transciphering framework for approximate homomorphic encryption," in *Advances in Cryptology-ASIACRYPT 2021*, M. Tibouchi and H. Wang, Eds., vol. 13092 of *Lecture Notes in Computer Science*, pp. 640–669, Springer, Singapore, 2021.

[11] L. Grassi, R. Lüftenegger, C. Rechberger, D. Rotaru, and M. Schofnegger, "On a generalization of substitution-permutation networks: the HADES design strategy," in *Advances in Cryptology-EUROCRYPT 2020*, A. Canteaut and Y. Ishai, Eds., vol. 12106 of *Lecture Notes in Computer Science*, pp. 674–704, Springer, Zagreb, Croatia, 2020.

[12] L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch, "Reinforced concrete: a fast hash function for verifiable computation," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022*, pp. 1323–1335, ACM, Los Angeles, CA, USA, 2022.

[13] J. Ha, S. Kim, B. Lee, J. Lee, and M. Son, "Rubato: noisy ciphers for approximate homomorphic encryption," in *Advances in Cryptology-EUROCRYPT. 2022*, O. Dunkelman and S. Dziembowski, Eds., vol. 13275 of *Lecture Notes in Computer Science*, pp. 581–610, Springer, Trondheim, Norway, 2022.

[14] T. Ashur, M. Mahzoun, and D. Toprakhisar, "Chaghri-a fhe-friendly block cipher," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 139–150, ACM, Los Angeles, CA, USA, 2022.

[15] C. Dobraunig, D. Kales, C. Rechberger, M. Schofnegger, and G. Zaverucha, "Shorter signatures based on tailor-made minimalist symmetric-key crypto," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022*, pp. 843–857, ACM, Los Angeles, CA, USA, 2022.

[16] L. Grassi, Y. Hao, C. Rechberger, M. Schofnegger, R. Walch, and Q. Wang, "Horst meets fluid-spn: griffin for zero-knowledge applications," in *Advances in Cryptology-CRYPTO. 2023*, H. Handschuh and A. Lysyanskaya, Eds., vol. 14083 of *Lecture Notes in Computer Science*, pp. 573–606, Springer, Santa Barbara, CA, USA, 2023.

[17] C. Bouvier, P. Briaud, P. Chaidos et al., "New design techniques for efficient arithmetization-oriented hash functions: ttanemoi permutations and ttjive compression mode," in *Advances in Cryptology-CRYPTO 2023*, H. Handschuh and A. Lysyanskaya, Eds., vol. 14083 of *Lecture Notes in Computer Science*, pp. 507–539, Springer, Santa Barbara, CA, USA, 2023.

[18] C. Dobraunig, L. Grassi, L. Helminger, C. Rechberger, M. Schofnegger, and R. Walch, "Pasta: a case for hybrid homomorphic encryption," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2023, no. 3, pp. 30–73, 2023.

[19] L. Grassi, M. Øygarden, M. Schofnegger, and R. Walch, "From farfalle to megafono via ciminion: the PRF hydra for MPC applications," in *Advances in Cryptology-EUROCRYPT 2023*, C. Hazay and M. Stam, Eds., vol. 14007 of *Lecture Notes in Computer Science*, pp. 255–286, Springer, Lyon, France, 2023.

[20] S. Kim, J. Ha, M. Son et al., "AIM: symmetric primitive for shorter signatures with stronger security," *Cryptology ePrint Archive-IACR*, Article ID 1387, 2022.

[21] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-knowledge from secure multiparty computation," in *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pp. 21–30, ACM, San Diego, California, USA, 2007.

[22] S. Duval, V. Lallemand, and Y. Rotella, "Cryptanalysis of the FLIP family of stream ciphers," in *Advances in Cryptology-CRYPTO 2016*, vol. 9814 of *Lecture Notes in Computer Science*, pp. 457–475, Springer, Santa Barbara, CA, USA, 2016.

[23] M. R. Albrecht, C. Cid, L. Grassi et al., "Algebraic cryptanalysis of stark-friendly designs: application to marvellous and MiMC," in *Advances in Cryptology-ASIACRYPT 2019*, S. D. Galbraith and S. Moriai, Eds., vol. 11923 of *Lecture Notes in Computer Science*, pp. 371–397, Springer, Kobe, Japan, 2019.

[24] F. Liu, T. Isobe, and W. Meier, "Cryptanalysis of full LowMC and LowMC-m with algebraic techniques," in *Advances in Cryptology-CRYPTO 2021*, T. Malkin and C. Peikert, Eds., vol. 12827 of *of Lecture Notes in Computer Science*, pp. 368–401, Springer, 2021.

[25] F. Liu, S. Sarkar, G. Wang, W. Meier, and T. Isobe, "Algebraic meet-in-the-middle attack on LowMC," in *Advances in Cryptology-ASIACRYPT 2022*, S. Agrawal and D. Lin, Eds., vol. 13791 of *Lecture Notes in Computer Science*, pp. 225–255, Springer, Taipei, Taiwan, 2022.

[26] I. Dinur, "Cryptanalytic applications of the polynomial method for solving multivariate equation systems over GF(2)," in *Advances in Cryptology-EUROCRYPT 2021*, A. Canteaut and F.-X. Standaert, Eds., vol. 12696 of *Lecture Notes in Computer Science*, pp. 374–403, Springer, Zagreb, Croatia, 2021.

[27] F. Liu, W. Meier, S. Sarkar, and T. Isobe, "New low-memory algebraic attacks on LowMC in the picnic setting," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 3, pp. 102–122, 2022.

[28] S. Banik, K. Barooti, F. B. Durak, and S. Vaudenay, "Cryptanalysis of LowMC instances using single plaintext/ciphertext pair," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 4, pp. 130–146, 2020.

[29] S. Banik, K. Barooti, S. Vaudenay, and H. Yan, "New attacks on LowMC instances with a single plaintext/ciphertext pair," in *Advances in Cryptology-ASIACRYPT 2021*, M. Tibouchi and H. Wang, Eds., vol. 13090 of *Lecture Notes in Computer Science*, pp. 303–331, Springer, Singapore, 2021.

[30] F. Liu, S. Sarkar, W. Meier, and T. Isobe, "Algebraic attacks on rasta and dasta using low-degree equations," in *Advances in Cryptology-ASIACRYPT 2021*, M. Tibouchi and H. Wang, Eds., vol. 13090 of *Lecture Notes in Computer Science*, pp. 214–240, Springer, Singapore, 2021.

[31] M. Eichlseder, L. Grassi, R. Lüftenegger et al., "An algebraic attack on ciphers with low-degree round functions: application to full MiMC," in *Advances in Cryptology-ASIACRYPT 2020*, S. Moriai and H. Wang, Eds., vol. 12491 of *Lecture Notes in Computer Science*, pp. 477–506, Springer, Daejeon, South Korea, 2020.

[32] T. Beyne, A. Canteaut, I. Dinur et al., "Out of oddity-new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems," in *Advances in Cryptology-CRYPTO 2020*, D. Micciancio and T. Ristenpart, Eds., vol. 12172 of *Lecture Notes in Computer Science*, pp. 299–328, Springer, Santa Barbara, CA, USA, 2020.

[33] C. Bouvier, A. Canteaut, and L. Perrin, "On the algebraic degree of iterated power functions," *Designs, Codes and Cryptography*, vol. 91, no. 3, pp. 997–1033, 2023.

[34] F. Liu, R. Anand, L. Wang, W. Meier, and T. Isobe, "Coefficient grouping: breaking chaghri and more," in *Advances in Cryptology-EUROCRYPT 2023*, C. Hazay and M. Stam, Eds., vol. 14007 of *Lecture Notes in Computer Science*, pp. 287–317, Springer, Lyon, France, 2023.

[35] F. Liu, L. Grassi, C. Bouvier, W. Meier, and T. Isobe, "Coefficient grouping for complex affine layers," in *Advances in Cryptology-CRYPTO 2023*, H. Handschuh and A. Lysyanskaya, Eds., vol. 14083 of *Lecture Notes in Computer Science*, pp. 540–572, Springer, Santa Barbara, CA, USA, 2023.

[36] L. Grassi, I. M. Ayala, M. N. Hovd, M. Øygarden, H. Raddum, and Q. Wang, "Cryptanalysis of symmetric primitives over rings and a key recovery attack on rubato," in *Advances in Cryptology-CRYPTO. 2023*, H. Handschuh and A. Lysyanskaya, Eds., vol. 14083 of *Lecture Notes in Computer Science*, pp. 305–339, Springer, Santa Barbara, CA, USA, 2023.

[37] K. Zhang, Q. Wang, Y. Yu, C. Guo, and H. Cui, "Algebraic attacks on round-reduced RAIN and full AIM-III," in *Advances in Cryptology–ASIACRYPT 2023*, vol. 14440 of *Lecture Notes in Computer Science*, pp. 285–310, Springer, Singapore, 2023.

[38] F. Liu, M. Mahzoun, M. Øygarden, and W. Meier, "Algebraic attacks on RAIN and AIM using equivalent representations," *IACR Transactions on Symmetric Cryptology*, vol. 2023, no. 4, pp. 166–186, 2023.

[39] F. Liu, A. Kalam, S. Sarkar, and W. Meier, "Algebraic attack on FHE-friendly cipher HERA using multiple collisions," *Cryptology ePrint Archive-IACR*, Article ID 1800, 2023.

[40] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Advances in Cryptology—CRYPTO '97. CRYPTO 1997*, B. S. Kaliski, Ed., vol. 1294 of *Lecture Notes in Computer Science*, pp. 513–525, Springer, Santa Barbara, California, USA, 1997.

[41] S. P. Skorobogatov and R. J. Anderson, "Optical fault induction attacks," in *Cryptographic Hardware and Embedded Systems-CHES 2002*, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds., vol. 2523 of *Lecture Notes in Computer Science*, pp. 2–12, Springer, Redwood Shores, CA, USA, 2002.

[42] J.-M. Schmidt and M. Hutter, "Optical and EM fault-attacks on CRT-based RSA: concrete results," 2007.

[43] Y. I. M. Keun Soo, "The rowhammer attack injection methodology," in *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, pp. 1–10, IEEE, Budapest, Hungary, 2016.

[44] S. Endo, T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "An on-chip glitchy-clock generator for testing fault injection attacks," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 265–270, 2011.

[45] L. Zussa, J.-M. Dutertre, J. Clédière, and A. Tria, "Power supply glitch induced faults on FPGA: an in-depth analysis of the injection mechanism," in *2013 IEEE 19th International On-Line Testing Symposium (IOLTS)*, pp. 110–115, IEEE, 2013.

[46] H. S. Lim, J. H. Lee, and D.-G. Han, "Novel fault injection attack without artificial trigger," *Applied Sciences*, vol. 10, no. 11, Article ID 3849, 2020.

[47] C. Roscian, J.-M. Dutertre, and A. Tria, "Frontside laser fault injection on cryptosystems-application to the AES' last round," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 119–124, IEEE, Austin, TX, USA, 2013.

[48] Z. Zhang, Y. Cheng, D. Liu, S. Nepal, and Z. Wang, "Telehammer: a formal model of implicit rowhammer," arXiv: Cryptography and Security, 2019.

[49] B. N. Bathe, S. Tiwari, R. Anand, D. Roy, and S. Maitra, "Differential fault attack on espresso," in *Progress in Cryptology-INDOCRYPT 2021*, A. Adhikari, R. Küsters, and B. Preneel, Eds., vol. 13143 of *Lecture Notes in Computer Science*, pp. 271–286, Springer, Jaipur, India, 2021.

[50] S. Maitra, A. Siddhanti, and S. Sarkar, "A differential fault attack on plantlet," *IEEE Transactions on Computers*, vol. 66, no. 10, pp. 1804–1808, 2017.

[51] S. Sarkar, P. Dey, A. Adhikari, and S. Maitra, "Probabilistic signature based generalized framework for differential fault

analysis of stream ciphers," *Cryptography and Communications*, vol. 9, no. 4, pp. 523–543, 2017.
[52] D. Roy, B. N. Bathe, and S. Maitra, "Differential fault attack on kreyvium & FLIP," *IEEE Transactions on Computers*, vol. 70, no. 12, pp. 2161–2167, 2021.
[53] R. Radheshwar, M. Kansal, P. Méaux, and D. Roy, "Differential fault attack on rasta and $\text{FiLIP}_{\text{DSM}}$," *IEEE Transactions on Computers*, vol. 72, no. 8, pp. 2418–2425, 2023.
[54] M. Dworkin, "SHA-3 standard: permutation-based hash and extendable-output functions," 2015.
[55] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, no. 4, pp. 354–356, 1969.
[56] M. Albrecht and G. Bard, "The m4ri library," 2021.
[57] J. Alman and V. V. Williams, "A refined laser method and faster matrix multiplication," ArXiv, abs/2010.05846, 2020.