Designs, Codes and Cryptography (2023) 91:2057–2075
https://doi.org/10.1007/s10623-022-01178-1

New cryptanalysis of LowMC with algebraic techniques

Wenxiao Qiao[1,2] · Hailun Yan[3] · Siwei Sun[3,4] · Lei Hu[1,2] · Jiwu Jing[3]

Received: 4 June 2022 / Revised: 26 October 2022 / Accepted: 22 December 2022 /
Published online: 17 February 2023

Abstract
LowMC is a family of block ciphers proposed by Albrecht et al. at EUROCRYPT 2015, which is tailored specifically for FHE and MPC applications. At ToSC 2018, a difference enumeration attack was given for the cryptanalysis of low-data instances of full LowMCv2 with few applied S-boxes per round. Recently at CRYPTO 2021, an efficient algebraic technique was proposed to attack 4-round LowMC adopting a full S-box layer. Following these works, we present a new difference enumeration attack framework, which is based on our new observations on the LowMC S-box, to analyze LowMC instances with a full S-box layer. As a result, with only 3 chosen plaintexts, we can attack 4-round LowMC instances which adopt a full S-box layer with block size of 129, 192, and 255 bits, respectively. We show that all these attacks have either a lower time complexity or a higher success probability than those reported in the CRYPTO paper.

Keywords LowMC · Algebraic attack · Linearization technique · 2-Difference

Mathematics Subject Classification 94A60

Communicated by S. D. Galbraith.

✉ Hailun Yan
hailun.yan@ucas.ac.cn

Wenxiao Qiao
qiaowenxiao@iie.ac.cn

Siwei Sun
sunsiwei@ucas.ac.cn

Lei Hu
hulei@iie.ac.cn

Jiwu Jing
jwjing@ucas.ac.cn

1 State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China

2 School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

3 School of Cryptology, University of Chinese Academy of Sciences, Beijing, China

4 State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

1 Introduction

The LowMC family of block ciphers [1] was first proposed by Albrecht et al. at EUROCRYPT 2015 and was designed to achieve low multiplicative complexity, which is tailored specifically for MPC [16, 18, 24, 29, 30] and FHE [7, 15] applications. LowMC uses flexible Substitution–Permutation-Network (SPN) constructions, where instantiations can be created by independently choosing the block size n, the key size k, the number of S-boxes m in the substitution layer and the allowed data complexity d of attacks. Especially, some of the instances adopt the so-called partial Substitution–Permutation Network (P-SPN), i.e. in which the S-boxes are applied over only partial state bits of the cipher.

LowMC has been utilized as the underlying block cipher of the post-quantum signature scheme Picnic [9], which is an alternative candidate in the third round NIST's Post-Quantum Cryptography competition [25]. Recently, alternative parameters of LowMC were chosen for Picnic3 [17]. Different from Picnic2 where a partial S-box layer is adopted when instantiating LowMC, a full S-box layer is used when generating the three instances of LowMC in Picnic3. In Picnic3, 4-round LowMC is recommended and 5-round LowMC is treated as an alternative.

The proposal of LowMC not only starts a new trend to design symmetric-key primitives, like FLIP [23], MiMC [2], Kreyvrium [8], Rasta [13], GMiMC [3], and Ciminion [14], but also raises new challenges for cryptanalysis to evaluate its security. Soon after its publication, a higher-order differential attack (ICISC 2015, Dobraunig et al. [12]) and an optimized interpolation attack (ASIACRYPT 2015, Dinur et al. [11]) were given, which directly made LowMC move to LowMC v2, although with a high data complexity. Later at FSE 2018, Rechberger et al. [27] proposed the so-called difference enumeration technique to analyse LowMC instances with a few S-boxes in each round. Rechberger et al.'s approach requires very little data—as little as 3 chosen plaintext–ciphertext pairs. To resist such attack, LowMC was further updated to LowMC v3.[1] At CRYPTO 2021, Liu et al. [19] revisited the difference enumeration technique for LowMC and showed that some important LowMC instances are still insecure. They achieved efficient key recovery attacks on 3 instances of 4-round LowMC, with only 2 chosen plaintexts. Recently, Liu et al. [22] proposed an algebraic meet-in-the-middle (MITM) technique to analyze LowMC with a partial S-box layer. As a result, the attacks on LowMC and LowMC-M [26] published at CRYPTO 2021 are further improved and some LowMC instances could be broken for the first time.

In another direction, cryptanalysis of the LowMC block cipher when the attacker has access to a single known plaintext/ciphertext pair is particularly relevant while arguing the security of the Picnic digital signature scheme. In Picnic, the plaintext/ciphertext pair generated by the LowMC block cipher serves as the public (verification) key and the corresponding LowMC encryption key also serves as the secret (signing) key of the signature scheme. Therefore, a data complexity one key recovery attack on LowMC block cipher will lead to a signature forgery on Picnic. Until now, there have been several attacks on LowMC in such scenario [4–6, 10, 20, 21]. At ToSC 2020, Banik et al. [4] proposed guess-and-determine attacks on reduced 2-round LowMC in the Picnic setting. Following this work, at ASIACRYPT 2021, Banik et al. [5] proposed 2-stage Meet-in-the-Middle (MITM) attack with gray-code based approach, which reduced the computational complexity of 2 rounds and extended the number of attacked round to 3 rounds. A parallel work [10] also shows that 2 out of 3 instances of the 4-round LowMC in the Picnic3 setting can be broken, but it requires a huge amount of memory. Later, Banik et al. [6] combine the linearization techniques of [4, 5] and the equation solving methods of [10] to analyse LowMC instances with complete non-linear

[1] For simplicity, LowMC represents LowMC v3 in the remaining part of this paper.

layers, which yields a drastic reduction in terms of memory complexity. At ToSC 2022, Liu et al. [21] significantly improve the attacks on LowMC in the Picnic Setting by using better time-memory tradeoffs. For a survey of key recovery attacks on LowMC in such attack scenario, readers may check the work done by Grassi et al. [31].

In this paper, we study the security of LowMC with low data complexity and we are most interested in Rechberger et al.'s work [27] and Liu et al.'s work [19]. In [27], Rechberger et al. presented a difference enumeration attack to analyse LowMC instances with a partial S-box layer. The difference enumeration attack is a chosen-plaintext attack, which consists of two steps. The first step is to encrypt a pair (or more) of chosen plaintexts and then recover the difference evolutions between the plaintexts through each component in each round with a meet-in-the-middle method, i.e. to recover the differential trail. This step is called the difference enumeration phase. The second step is to derive the secret key from the recovered differential trail. This step is called the key-recovery phase. As a result, the number of the required plaintexts can be as small as 4. Furthermore, the authors showed that it is more effective to consider d-differences instead of simple differences.

However, the original difference enumeration attack [27] doesn't fit well with LowMC instances with a full S-box layer. At CRYPTO 2021, Liu et al. [19] showed a new difference enumeration attack framework to attack the constructions adopting a full S-box layer with 2 chosen plaintexts.

1.1 Our contributions

We propose a new difference enumeration attack framework for LowMC instances with a full S-box layer. Instead of considering the traditional difference, we turn to consider the 2-difference and give some new observations on the LowMC S-box which can be exploited in our attack. We then enumerate 2-differences with algebraic techniques and efficiently derive the master key from the recovered 2-differential trails with the linearization technique. Finally, we apply our attack framework to 4-round LowMC with block size of 129, 192, 255 bits, respectively. Our results are summarized in Table 1. Our attacks have a quite low data complexity, which is only 3 chosen plaintexts. And all these attacks have either a lower time complexity or a higher success probability than those reported by Liu et al. in the previous CRYPTO paper.

1.2 Organization of the paper

We give a brief introduction of LowMC and some definitions in Sect. 2. In Sect. 3, we revisit the difference enumeration techniques. In Sect. 4, we introduce our approach in a high level and show some new observations on LowMC S-box. In Sect. 5, we introduce how to find all valid 4-round compact 2-differential trials in our attack by solving linear equations. In Sect. 6, we show how to recover the master key with the algebraic method. The analysis and experimental results of our attack on 4-round LowMC instances are given in Sect. 7. Finally, we conclude the paper in Sect. 8.

Table 1 A summary of the results for 4-round LowMC instances with a full S-box layer, where the time complexity is estimated in units which equal to a single encryption operation of the relevant 4-round LowMC

n	k	m	Data	Time	Memory	Success Pro.	References
129	129	43	3	2^{123}	Negligible	0.83	This paper
			2	$2^{122.6}$	Negligible	0.80[a]	[19]
			2	2^{104}	Negligible	0.24	[19]
192	192	64	3	$2^{185.6}$	Negligible	0.994	This paper
			2	$2^{187.6}$	Negligible	0.99	[19]
			2	2^{180}	Negligible	0.82	[19]
			2	2^{156}	Negligible	0.247	[19]
255	255	85	3	2^{243}	Negligible	0.994	This paper
			3	2^{242}	Negligible	0.989	This paper
			2	$2^{246.6}$	Negligible	0.986	[19]
			2	$2^{236.6}$	Negligible	0.848	[19]
			2	2^{208}	Negligible	0.2465	[19]

For the case of $(n, k, m) = (129, 129, 43)$, our attack has a higher success probability; for $(n, k, m) = (192, 192, 64)$, our attack is 2^2 times faster than that proposed in [19] when limiting the success probability to 0.99 or more; for $(n, k, m) = (255, 255, 85)$, our attack is $2^{4.6}$ times faster than that proposed in [19] when limiting the success probability to 0.986 or more

[a] Success probability recalculated is higher than that reported in [19]

2 Preliminaries

2.1 A brief description of LowMC

LowMC [1] is a family of block ciphers with flexible SPN constructions. When instantiating LowMC, users can independently choose the parameters: the block size n, the key size k, the number of S-boxes m in each round and the allowed data complexity 2^D of attacks. The number of rounds R needed to reach the security against several known attacks with reasonable security margins is then derived from these parameters. The block cipher uses a 3-bit S-box which is the only non-linear transformation in the construction. Both the linear layers and the round key generation are done by multiplying with full rank matrices over GF(2) of appropriate dimensions.

The encryption procedure of LowMC starts with a key whitening (WK), and then iterates the round function (as depicted in Fig. 1) R times, which consists of four operations in the following order.

1. SBoxLayer(SB): A 3-bit S-box $(y_0, y_1, y_2) = S(x_0, x_1, x_2)$ with $(y_0, y_1, y_2) = (x_0 \oplus x_1x_2, x_0 \oplus x_1 \oplus x_0x_2, x_0 \oplus x_1 \oplus x_2 \oplus x_0x_1)$ is applied to the first $3m$ bits of the state in parallel. For the remaining $n - 3m$ bits, an identity mapping is applied.
2. LinearLayer(L): The n-bit state is multiplied with an invertible $n \times n$ matrix L_i in $GF(2)$. The matrix L_i is randomly chosen from all invertible binary $n \times n$ matrices.
3. ConstantAddition(AC): The n-bit state is XORed with an n-bit binary round constant RC_i. The round constant RC_i is randomly chosen from all binary vectors of length n.

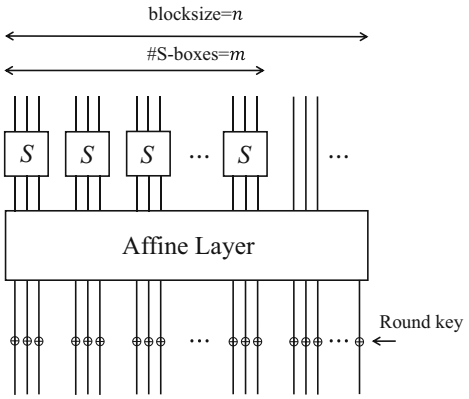

Fig. 1 The LowMC Round Function

4. KeyAddition(AK): The n-bit state is XORed with the n-bit round key K_{i+1}. To generate K_{i+1}, a matrix U_{i+1} is randomly chosen from all full rank $n \times k$ binary matrices, and then the K_{i+1} is obtained by multiplying the k-bit master key with U_{i+1}.

The whitening key K_0 is also calculated by multiplying the master key with a random full-rank $n \times k$ binary matrix U_0.

2.2 Definitions

Definition 1 (*d-Difference* [28]) For a tuple of $(d+1)$ texts $(s_0, s_1, \ldots, s_d)$, the corresponding d-difference is defined as $(\Delta^1, \Delta^2, \cdots, \Delta^d) = (s_0 \oplus s_1, s_0 \oplus s_2, \ldots, s_0 \oplus s_d)$. And the reference text s_0 is called the anchor of the d-difference.

We denote the plaintext by p and the ciphertext by c. The state after WK is denoted by X_0. In the i-th round, the input state of SB is denoted by X_i and the output state of SB is denoted by X_{i_S}, as shown below:

$$p \xrightarrow{WK} X_0 \xrightarrow{SB} X_{0\_S} \xrightarrow{L} \xrightarrow{AC} \xrightarrow{AK} X_1 \cdots X_{R-1} \xrightarrow{SB} X_{(R-1)\_S} \xrightarrow{L} \xrightarrow{AC} \xrightarrow{AK} c.$$

In particular, the 1-difference of plaintexts is denoted by Δ_p. In the i-th round, we denote the 1-difference of the input state of SB by Δ_i, and the 1-difference of the output state of SB by Δ_{i_S}, as shown below:

$$\Delta_p \xrightarrow{WK} \Delta_0 \xrightarrow{SB} \Delta_{0\_S} \xrightarrow{L} \xrightarrow{AC} \xrightarrow{AK} \Delta_1 \cdots \Delta_{R-1} \xrightarrow{SB} \Delta_{(R-1)\_S} \xrightarrow{L} \xrightarrow{AC} \xrightarrow{AK} \Delta_R.$$

Definition 2 (*Compact 1-Differential Trail* [19]) Let $\Delta_0 \to \Delta_1 \to \cdots \to \Delta_r$ be a 1-differential trail, in which we may not know all $\Delta_i (0 \le i \le r)$. If all (Δ_j, Δ_{j_S}) $(0 \le j \le r-1)$ and Δ_r are known, we call it an r-round compact 1-differential trail.

Definition 3 (*Compact d-Differential Trail* $(d > 1)$) Let $\alpha_0 \to \alpha_{0\_S} \to \alpha_1 \to \cdots \to \alpha_r$ where $\alpha_i = (\Delta_i^1, \Delta_i^2, \cdots, \Delta_i^d)$ with $(0 \le i \le r)$ and $\alpha_{j\_S} = (\Delta_{j\_S}^1, \Delta_{j\_S}^2, \cdots, \Delta_{j\_S}^d)$ with $(0 \le j \le r-1)$ be a d-differential trail, in which we may not know all (α_j, α_{j_S}) $(0 \le j \le r-1)$ and α_r. If all (α_j, α_{j_S}) $(0 \le j \le r-1)$ and α_r are known, we call it an r-round compact d-differential trail.

3 The difference enumeration attack framework

In this section, we briefly revisit the original difference enumeration attack [27] on instances with a partial S-box layer and the extended difference enumeration attack [19] on instances with a full S-box layer.

At ToSC 2018, the LowMC team proposed a difference enumeration attack [27] to analyze the security of LowMCv2 with a low data complexity. The difference enumeration attack consists of two phases. The first phase is called the difference enumeration phase, which is to recover internal d-differences for a chosen $(d+1)$-tuple of plaintexts and the corresponding ciphertexts. In this phase, a meet-in-the-middle approach is applied. The second phase is the key-recovery phase, which is to derive the secret key from the recovered compact d-differential trail.

However, the original difference enumeration attack [27] is not quite efficient when it comes to a full S-box layer. To refine the original difference enumeration attack, in the difference enumeration phase, Liu et al. [19] consider to choose a desirable input difference such that it will activate as few S-boxes as possible in the first two rounds. Moreover, they consider to enumerate the solutions of a linear equation system. In the key-recovery phase, for a retrieved 4-round compact 1-differential trail, they recover the full key by solving linear equations with k-bit master key and some internal variables.

The algebraic techniques used in this extended attack are based on the following observations on LowMC S-box.

Observation 1 [19] For each valid non-zero 1-difference transition $(\Delta x_0, \Delta x_1, \Delta x_2) \rightarrow (\Delta y_0, \Delta y_1, \Delta y_2)$, the inputs conforming to such a difference transition will form an affine space of dimension 1. In addition, (y_0, y_1, y_2) becomes linear in (x_0, x_1, x_2). A similar property also applies to the inverse of the S-box.

Observation 2 [19] For each non-zero input 1-difference $(\Delta x_0, \Delta x_1, \Delta x_2)$, its valid output 1-differences form an affine space of dimension 2. A similar property also applies to the inverse of the S-box.

Observation 3 [19] For an inactive S-box, the input becomes linear in the output after guessing two output bits, and the output becomes linear in the input after guessing two input bits. The same property holds for its inverse.

4 Approach overview and new observations on the LowMC S-box

In this section, we give an overview of our new difference enumeration attack on LowMC with a full S-box layer, and show our new observations on LowMC S-box.

4.1 Overview of our approach

In our new difference enumeration attack, we consider the 2-difference, and we call it 2-difference enumeration attack in the following. It also consists of two phases, i.e. the 2-difference enumeration phase and the key-recovery phase.

First, for the construction with a full S-box layer, the cost of enumerating d-differences in the original difference enumeration attack [27] is rather high, especially when $d > 1$. And if we enumerate 2-differences for more than one round, the time complexity will be

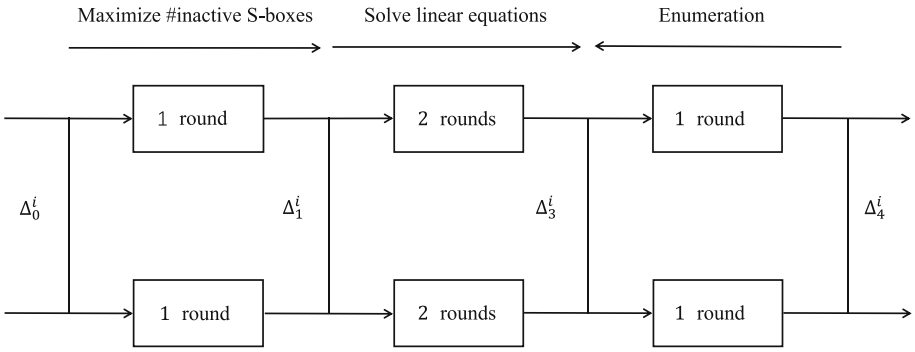

Fig. 2 The framework of the 2-difference enumeration attack

higher than that of the brute force attack. In order to overcome this obstacle, we choose a desirable input 2-difference such that the number of inactive S-boxes in the 0th round is maximized, as depicted in Fig. 2. Moreover, based on the algebraic techniques used in [19], we introduce some variables to represent internal 2-differences, and then construct and solve linear equations to find the valid 2-differences in the middle 2 rounds.

Second, for a recovered 4-round compact 2-differential trail, we can derive the master key by some algebraic techniques. Specifically, by exploiting the special property of the LowMC S-box, we can linearize the S-box. And if a S-box is active, the input and output will satisfy some linear equations. Finally, we can obtain a linear equation system in terms of the master key and some internal variables. Each solution of this equation system corresponds to a candidate master key, and check its correctness via a plaintext-ciphertext pair.

4.2 New observations on the LowMC S-box

Before introducing the details of our attacks on LowMC, it is necessary to describe our new observations on the LowMC S-box with respect to 2-differences. Denote a tuple of 3 input states of the S-box by (X_0, X_1, X_2) and the corresponding 6-bit input 2-difference $(X_0 \oplus X_1, X_0 \oplus X_2)$ of which the anchor is X_0 by $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1, \Delta x_0^2, \Delta x_1^2, \Delta x_2^2)$. Also, denote a tuple of 3 output states of the S-box by (Y_0, Y_1, Y_2) where $Y_i = S(X_i)(0 \leq i \leq 2)$ and the corresponding 6-bit output 2-difference $(Y_0 \oplus Y_1, Y_0 \oplus Y_2)$ of which the anchor is Y_0 by $(\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$.

Observation 4 For each non-zero input 2-difference $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1, \Delta x_0^2, \Delta x_1^2, \Delta x_2^2)$:

1. if $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (\Delta x_0^2, \Delta x_1^2, \Delta x_2^2)$, or $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (0, 0, 0)$, or $(\Delta x_0^2, \Delta x_1^2, \Delta x_2^2) = (0, 0, 0)$, its valid output 2-differences will form an affine space of dimension 2.
2. else, its valid output 2-differences will form an affine space of dimension 3.

A similar property also applies to the inverse of the S-box.

Example When the input 2-difference is $(0, 0, 1, 0, 0, 1)$, the corresponding output 2-difference $(\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$ satisfies

$$\begin{cases} \Delta y_0^1 \oplus \Delta y_0^2 = 0, \\ \Delta y_1^1 \oplus \Delta y_1^2 = 0, \\ \Delta y_2^1 \oplus \Delta y_2^2 = 0, \\ \Delta y_2^1 = 1. \end{cases}$$

Then the corresponding valid output 2-differences form an affine space of dimension 2.

When the input 2-difference is $(0, 0, 1, 1, 1, 0)$, the corresponding output 2-difference $(\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$ satisfies

$$\begin{cases} \Delta y_0^2 \oplus \Delta y_1^2 = 1, \\ \Delta y_0^1 \oplus \Delta y_1^1 \oplus \Delta y_2^2 = 1, \\ \Delta y_2^1 = 1. \end{cases}$$

Then the corresponding valid output 2-differences form an affine space of dimension 3.

Definition 4 For each valid **non-zero** 2-difference transition $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1, \Delta x_0^2, \Delta x_1^2, \Delta x_2^2) \rightarrow (\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$:

1. if $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (\Delta x_0^2, \Delta x_1^2, \Delta x_2^2)$, or $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (0, 0, 0)$, or $(\Delta x_0^2, \Delta x_1^2, \Delta x_2^2) = (0, 0, 0)$, we call the S-box **special-active (with respect to the 2-difference)**;
2. else, we call the S-box **non-special-active (with respect to the 2-difference)**.

For the 2-difference transition $(0, 0, 0, 0, 0, 0) \rightarrow (0, 0, 0, 0, 0, 0)$, we call the S-box **inactive (with respect to the 2-difference)**.

Observation 5 For each valid non-zero 2-difference transition $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1, \Delta x_0^2, \Delta x_1^2, \Delta x_2^2) \rightarrow (\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$, if the S-box is non-special-active (with respect to the 2-difference), the value of anchor X_0 will be determined.

Example If the 2-difference transition $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1, \Delta x_0^2, \Delta x_1^2, \Delta x_2^2) \rightarrow (\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)$ is $(0, 1, 1, 1, 0, 1) \rightarrow (0, 1, 0, 0, 0, 1)$, the value of anchor X_0 will be determined, i.e. $X_0 = (0, 1, 0)$.

4.2.1 Generalization

The above Observations 4 and 5 hold for all 3-bit almost perfect nonlinear (APN) S-boxes. As for Observation 5, this generalization is trivial. As for Observation 4, a simplified proof for this generalization can be referred to Appendix 1.

5 2-Difference enumeration

In this section, we first introduce how to enumerate 2-differences in the middle 2 rounds by solving linear equations, and then describe the procedure of 2-difference enumeration phase in our attack. Since we only consider 2-differences in the following, we will omit the phrase "with respect to the 2-difference" for simplicity.

5.1 Enumerating 2-differences via solving equations

Assume that $\alpha_i = (\Delta_i^1, \Delta_i^2)$ and $\alpha_{(i+1)\_S} = (\Delta_{(i+1)\_S}^1, \Delta_{(i+1)\_S}^2)$ are known in the i-th round and $(i+1)$-th round. We aim to enumerate all values of $\alpha_{i\_S} = (\Delta_{i\_S}^1, \Delta_{i\_S}^2)$ such that the 2-difference transition $\alpha_i \to \alpha_{i\_S} \to \alpha_{(i+1)\_S}$ is valid. Consider the general case: there are a inactive S-boxes and b special-active S-boxes in the i-th round, and there are c inactive S-boxes and d special-active S-boxes in the $(i+1)$-th round.

First, we introduce some variables to represent internal 2-differences in the i-th round. For the input 2-difference α_i, we can introduce at most $6m$ variables to represent the $6m$-bit output 2-difference $\alpha_{i\_S} = (\Delta_{i\_S}^1, \Delta_{i\_S}^2)$. However, by exploiting Observation 4, we could introduce $3m-3a-b$ variables to represent $\alpha_{i\_S} = (\Delta_{i\_S}^1, \Delta_{i\_S}^2)$. Specifically, 1) for an inactive S-box, the output 2-difference is determined, i.e. $(0, 0, 0, 0, 0, 0)$, so there is no need to introduce variables to represent them. 2) For a special-active S-box, the valid output 2-differences form an affine space of dimension 2, so we need to introduce 2 variables to represent them. Thus, we need introduce $2b$ variables $(v_0, v_1, \ldots, v_{2b-1})$ to represent the output 2-differences of the b special-active S-boxes. 3) For a non-special-active S-box, the valid output 2-differences form an affine space of dimension 3, so we need to introduce 3 variables to represent them. Thus, we need introduce $3(m-a-b)$ variables $(v_{2b}, v_{2b+1}, \ldots, v_{2b+3(m-a-b)-1})$ to represent the output 2-differences of the $(m-a-b)$ non-special-active S-boxes. As a result, we only need to introduce $2b+3(m-a-b) = 3m-3a-b$ variables $(v_0, v_1, \ldots, v_{3m-3a-b-1})$ to denote the valid output 2-difference α_{i_S}. In this way, each bit of $\alpha_{i+1} = (\Delta_{i+1}^1, \Delta_{i+1}^2)$ can be written as a linear expression with variables $(v_0, v_1, \ldots, v_{3m-3a-b-1})$.

Then, in the $(i+1)$-th round, the output 2-difference $\alpha_{(i+1)\_S}$ is known, so we can construct an equation system with the above variables $(v_0, v_1, \ldots, v_{3m-3a-b-1})$ based on Observation 4. Specifically, (1) for an inactive S-box, the input 2-difference is $(0, 0, 0, 0, 0, 0)$, i.e. in α_{i+1} the values of 6 bits which are linear in the above variables are known. Thus, six linear equations with the above variables can be obtained. (2) For a special-active S-box, its valid input 2-differences form an affine space of dimension 2, i.e. the value of 6-bit input 2-difference satisfies 4 linear equations. Thus, four linear equations with the above variables can be obtained. (3) For a non-special-active S-box, its valid input 2-differences form an affine space of dimension 3, i.e. the value of 6-bit input 2-difference satisfies 3 linear equations. Thus, three linear equations with the above variables can be obtained. Therefore, we can obtain $6c+4d+3(m-c-d) = 3m+3c+d$ linear equations with the above $3m-3a-b$ variables. Since $3m-3a-b \le 3m \le 3m+3c+d$, we can expect the equation system has at most one solution. And the solution will correspond to a valid value of α_{i_S}.

5.1.1 Complexity evaluation

The time complexity of solving the above $3m+3c+d$ linear equations with $3m-3a-b$ variables is estimated as n^3+2n^2 bit operations. Specifically, we first solve $3m-3a-b$ linear equations among them by Gaussian elimination (GE), which costs around n^3 bit operations. And we can expect the number of solutions is one. Then we check the correctness of this solution by the remaining $(3m+3c+d)-(3m-3a-b)$ linear equations, which costs $((3m+3c+d)-(3m-3a-b))(3m-3a-b) \le 2n^2$ bit operations. Since performing a LowMC encryption costs around $2n^2R$ bit operations, the time complexity of solving the above equation system is equivalent to $\frac{n^3+2n^2}{2n^2R} \approx \frac{n^3}{2n^2R}$ encryptions.

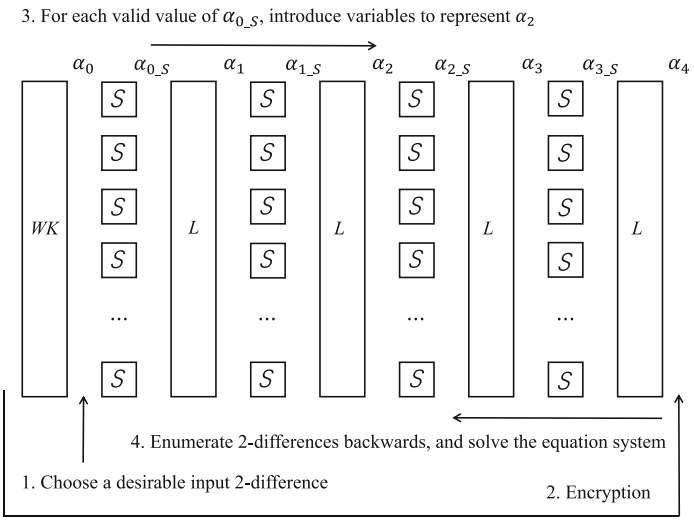

Fig. 3 The procedure of the 2-difference enumeration phase

5.2 Recovering valid 2-differential trails

Now we introduce the 2-difference enumeration phase of our attack on 4-round LowMC with a full S-box layer in detail. As depicted in Fig. 3, our 2-difference enumeration phase consists of the following 4 steps:

1. We choose a desirable input 2-difference α_0 such that there are 1 non-special-active S-box and $m-1$ inactive S-boxes in the 0th round.
2. Encrypt 3 plaintexts (p_0, p_1, p_2) whose 2-difference is α_0, and obtain the corresponding ciphertexts (c_0, c_1, c_2). Then we compute the 2-difference α_4 of (c_0, c_1, c_2) and α_{3_S}.
3. For each of 8 possible values of α_{0_S}, we compute the value of α_1 from α_{0_S} and introduce $3m-3a-b$ variables $(v_0, v_1, \ldots, v_{3m-3a-b-1})$ to represent α_2 using the method described in 5.1, where there are a inactive S-boxes and b special-active S-boxes in the 1st round. Then go to the next step.
4. Enumerate 2-differences backwards for 1 round from α_{3_S} to α_{2_S}. According to each value of α_{2_S}, by the method described in 5.1, we obtain $3m+3c+d$ linear equations with the $3m-3a-b$ variables $(v_0, v_1, \ldots, v_{3m-3a-b-1})$, where there are c inactive S-boxes and d special-active S-boxes in the 2nd round. Solve this equation system and we can expect it has at most one solution. For each solution, a valid 4-round compact 2-differential trail is found.

5.2.1 Complexity evaluation

As in [19], we compute the expected number of iterations to enumerate the 2-differences backwards in the 2-difference enumeration phase. In our attack using 3 chosen plaintexts, α_{3_S} is a random fixed value. We assume that there are t inactive S-boxes and j special-active S-boxes in the 3rd round. In this phase, for each possible value of α_{0_S}, the expected number of iterations to enumerate the 2-differences backwards is

$$T_0 = 4^j \times 8^{m-t-j}$$

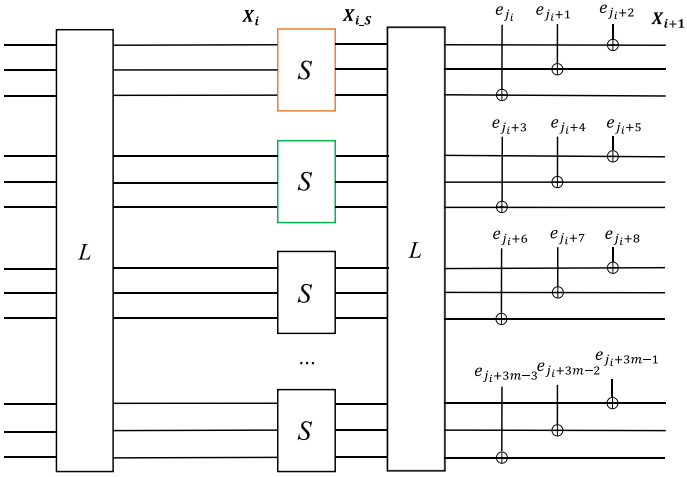

Fig. 4 Introduce extra variables and extract linear equations in the i-th round ($1 \leq i \leq 3$), where inactive S-boxes are colored in orange, special-active S-boxes are colored in green and non-special-active S-boxes are colored in black. Note that $j_i = (9-3i)m$

$$= 2^{3m-3t-j}. \tag{1}$$

Thus the expected number of iterations to enumerate the 2-differences backwards in total is

$$\begin{aligned} T_1 &= 8 \times 4^j \times 8^{m-t-j} \\ &= 2^{3m-3t-j+3}. \end{aligned} \tag{2}$$

As in 5.1, we simply estimate the cost of solving the equation system as $\frac{n^3}{2n^2R} = \frac{n}{2R}$ when enumerating 2-differences backwards each time. Thus, the expected time complexity of solving the equation systems in this phase is

$$T_2 = 2^{3m-3t-j+3} \times \frac{n}{2R}. \tag{3}$$

Therefore, the expected time complexity of the 2-difference enumeration phase is

$$\begin{aligned} T_d &= T_1 + T_2 \\ &\approx 2^{3m-3t-j+3} \times \frac{n}{2R}. \end{aligned} \tag{4}$$

6 Key recovery with algebraic techniques

In this section, we show the algebraic techniques which are used to derive the full key when a 4-round compact 2-differential trail is recovered in our attack. For each compact 2-differential trail we find, there are 1 non-special-active S-box and $m-1$ inactive S-boxes in the 0th round. And we consider the general case: there are a inactive S-boxes and b special-active S-boxes in the 1st round, c inactive S-boxes and d special-active S-boxes in the 2nd round, and t inactive S-boxes and j special-active S-boxes in the 3rd round.

Now we consider the encryption path from p_0 to c_0. The procedure starts from the 3rd round and can be divided into the following steps (as depicted in Fig. 4):

1. Denote the round key bits used in the 3rd round by $(e_0, e_1, \ldots, e_{3m-1})$. Then X_{3_S} becomes linear in $(e_0, \ldots, e_{3m-1})$. For the t inactive S-boxes, introduce extra $3t$ variables $(v_0, v_1, \ldots, v_{3t-1})$ to represent their input bits. Based on Observation 1, for the j special-active S-boxes, we obtain $2j$ linear equations with the output bits of these S-boxes, and the input bits become linear in the output bits for these S-boxes. Based on Observation 5, for the $m-t-j$ non-special-active S-boxes, we obtain $3(m-t-j)$ linear equations with the output bits of these S-boxes, and the input bits of these S-boxes are determined. Then we obtain $2j + 3(m-t-j) = 3m - 3t - j$ linear equations with $(e_0, e_1, \ldots, e_{3m-1})$ and X_3 becomes linear in $(v_0, v_1, \ldots, v_{3t-1}, e_0, e_1, \ldots, e_{3m-1})$.
2. Move to the 2nd round, and denote the round key bits used in this round by $(e_{3m}, e_{3m+1}, \ldots, e_{6m-1})$. Then X_{2_S} becomes linear in $(v_0, v_1, \ldots, v_{3t-1}, e_0, \ldots, e_{6m-1})$. For each inactive S-box, we guess 2 bits for its output and then its input bits become linear in the output bits according to Observation 3. From the 2 guessed bits, we obtain 2 linear equations with the output bits of the inactive S-box. Similarly to that in the 3rd round, for each special-active S-box, we obtain 2 linear equations with its output bits and its input bits become linear in the output bits. For each non-special-active S-box, we obtain 3 linear equations with its output bits and its input bits are determined. Then we obtain $2c + 2d + 3(m-c-d) \geq 2m$ linear equations with $(v_0, v_1, \ldots, v_{3t-1}, e_0, \ldots, e_{6m-1})$ and X_2 becomes linear in these variables.
3. Move to the 1st round, and denote the round key bits used in this round by $(e_{6m}, e_{6m+1}, \ldots, e_{9m-1})$. Then X_{1_S} becomes linear in $(v_0, v_1, \ldots, v_{3t-1}, e_0, \ldots, e_{9m-1})$. Similarly to that in the 2nd round and 3rd round, for the special-active S-boxes and non-special-active S-boxes, we obtain $2b + 3(m-a-b) = 3m - 3a - b$ linear equations with $(v_0, v_1, \ldots, v_{3t-1}, e_0, \ldots, e_{9m-1})$.
4. Move to the 0th round. For the non-special-active S-box, there are 3 linear equations in terms of the plaintext and the whitening key.
5. Since each round key bit is linear in the k-bit master key, we obtain $(3m-3t-j)+2m+(3m-3a-b)+3 = 8m+3-3a-b-3t-j$ linear equations with $(v_0, v_1, \ldots, v_{3t-1})$ and k-bit master key. For each solution of the equation system, we get a candidate master key and check it via the plaintext-ciphertext pair. Then try another guess in the 2nd round and repeat the same procedure until all possible guesses are traversed.

6.1 Complexity evaluation

Assume that there are t inactive S-boxes and j special-active S-boxes in the 3rd round. For the valid value of α_{0_S} with a inactive S-boxes and b special-active S-boxes in the 1st round, when recovering a compact 2-differential trail each time, we derive the master key by constructing $8m+3-3a-b-3t-j$ linear equations with $k+3t$ variables after guessing some bits. If $k+3t \leq 8m+3-3a-b-3t-j$, i.e. $6t+j \leq 5m+3-3a-b$, then it can be expected that the equation system has at most 1 solution. The expected time complexity of retrieving the master key for this case is

$$\begin{aligned} T_3 &= \sum_{c=0}^{m}\sum_{d=0}^{m-c} 4^b \times 8^{m-a-b} \times 4^j \times 8^{m-t-j} \times \binom{m}{c}\binom{m-c}{d}\left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d \\ &\quad \times \left(\frac{42}{64}\right)^{m-c-d} \times 4^d \times 8^{m-c-d} \times 2^{-2n} \times 2^{2c} \\ &\leq 2^{2.73m-3t-j-3a-b}. \end{aligned} \tag{5}$$

If $k+3t > 8m+3-3a-b-3t-j$, i.e. $6t+j > 5m+3-3a-b$, then it can be expected that the equation system has $2^{6t+j-5m-3+3a+b}$ solutions. The expected time complexity of retrieving the master key for this case is

$$\begin{aligned} T_3 &= \sum_{c=0}^{m}\sum_{d=0}^{m-c} 4^b \times 8^{m-a-b} \times 4^j \times 8^{m-t-j} \times \binom{m}{c}\binom{m-c}{d}\left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d \\ &\quad \times \left(\frac{42}{64}\right)^{m-c-d} \times 4^d \times 8^{m-c-d} \times 2^{-2n} \times 2^{2c} \times 2^{6t+j-5m-3+3a+b} \\ &\leq 2^{3t-2.27m-3}. \end{aligned} \tag{6}$$

A detailed explanation for the complexity T_3 can be referred to Appendix 2.

7 Attacks on LowMC with a full S-box layer

Combining the two phases in Sects. 5 and 6, we now apply our attack framework (as shown in Algorithms 1 and 2) to analyse three 4-round LowMC instances with parameters $(n, k, m, D) \in \{(129, 129, 43, 2), (192, 192, 64, 2), (255, 255, 85, 2)\}$. The results are summarized in Table 1.

Algorithm 1: 2-Difference Enumeration Attack

Input: Encryption oracle of a 4-round LowMC instance with parameter (n, k, m).

Output: The master key.

```
1 Choose a input 2-difference α_0 which has 1 non-special-active S-box and m − 1 inactive S-boxes.
2 Ask the encryption oracle to provide the encryption of (p_0, p_1, p_2) whose 2-difference equals α_0.
  Obtain the corresponding ciphertexts (c_0, c_1, c_2) and compute α_{3_S}.
3 for valid value of α_{0_S} do
4     Introduce variables to represent α_2.
5     for α_{2_S} enumerated backwards do
6         Construct and solve linear equations.
7         for solution obtained do
8             Come to Algorithm 2.
```

Now we calculate the time complexity and success probability of our attack, which needs a negligible memory. In our attack with 3 chosen plaintexts, α_{3_S} is a random fixed value. For each S-box in the 3rd round, the probability that this S-box is inactive is $\frac{1}{64}$, the probability that this S-box is special-active is $\frac{21}{64}$, and the probability that this S-box is non-special-active is $\frac{42}{64}$. In the following, we denote the probability of event w happening by $\Pr[w]$.

Attack on $(129, 129, 43, 2)$**.** For $(n, k, m, D, R) = (129, 129, 43, 2, 4)$, as $\Pr[3t + j \geq 13] \approx 0.83$, we conclude that with the success probability 0.83, the total time complexity to enumerate 2-differences will be $2^{3m+3-3t-j} \times \frac{n}{2R} \leq 2^{123}$ based on Eq. 4, and the total time complexity to retrieve the master key will not exceed $8 \times max\{2^{2.73m-3t-j}, 2^{3t-2.27m-3}\} \leq 2^{107.4}$ based on Eqs. 5 and 6. Thus, we can break the parameter $(n, k, m, D, R) = (129, 129, 43, 2, 4)$ with time complexity less than 2^{123} and success probability 0.83.

Attack on $(192, 192, 64, 2)$**.** For $(n, k, m, D, R) = (192, 192, 64, 2, 4)$, as $\Pr[3t + j \geq 14] \approx 0.994$, we conclude that with success probability 0.994, the total time complexity to

Algorithm 2: Derive Master Key from a Compact 2-Differential Trail

Input: A 4-round compact 2-differential trail.

Output: The master key.

```
for S-box in the 3rd round do
    if S-box is inactive then
        Introduce 3 variables to represent its input.
    if S-box is special-active then
        Obtain 2 linear equations with its output.
    if S-box is non-special-active then
        Obtain 3 linear equations with its output.
Move to the 2nd round.
for value of guessed output bits of the inactive S-boxes do
    Obtain 2m linear equations in total with the output of the m S-boxes.
    for S-box in the 1st round do
        if S-box is special-active then
            Obtain 2 linear equations with its output.
        if S-box is non-special-active then
            Obtain 3 linear equations with its output.
    Move to the 0th round and obtain 3 linear equations with the input of the non-special-active S-box.
    Solve the linear equations.
    for solution obtained do
        Check it via a plaintext-ciphertext pair.
        if passes the verification then
            return the solution.
```

enumerate 2-differences will be $2^{3m+3-3t-j} \times \frac{n}{2R} \leq 2^{185.6}$ based on Eq. 4, and the total time complexity to retrieve the master key will not exceed $8 \times max\{2^{2.73m-3t-j}, 2^{3t-2.27m-3}\} \leq 2^{163.7}$ based on Eqs. 5 and 6. Thus, we can break the parameter $(n, k, m, D, R) = (192, 192, 64, 2, 4)$ with time complexity less than $2^{185.6}$ and success probability 0.994.

Attack on $(255, 255, 85, 2)$**.** For $(n, k, m, D, R) = (255, 255, 85, 2, 4)$, as $\Pr[3t + j \geq 20] \approx 0.994$, we conclude that with success probability 0.994, the total time complexity to enumerate 2-differences will be $2^{3m+3-3t-j} \times \frac{n}{2R} \leq 2^{243}$ based on Eq. 4, and the total time complexity to retrieve the master key will not exceed $8 \times max\{2^{2.73m-3t-j}, 2^{3t-2.27m-3}\} \leq 2^{215}$ based on Eqs. 5 and 6. Thus, we can break the parameter $(n, k, m, D, R) = (255, 255, 85, 2, 4)$ with time complexity less than 2^{243} and success probability 0.994.

In addition, as $\Pr[3t + j \geq 21] \approx 0.989$, we conclude that with success probability 0.989, the total time complexity to enumerate 2-differences will be $2^{3m+3-3t-j} \times \frac{n}{2R} \leq 2^{242}$ based on Eq. 4, and the total time complexity to retrieve the master key will not exceed $8 \times max\{2^{2.73m-3t-j}, 2^{3t-2.27m-3}\} \leq 2^{214}$ based on Eqs. 5 and 6. Thus, we can break the parameter $(n, k, m, D, R) = (255, 255, 85, 2, 4)$ with time complexity less than 2^{242} and success probability 0.989.

7.1 Experiments

In order to confirm the correctness of our methods, similarly to that in [19], we performed experiments on the toy LowMC instance with parameter $(n, k, m, D, R) = (21, 21, 7, 2, 4)$. We provide our code at https://github.com/wxqiao/LowMC_new_attack_2diff.

By choosing different desirable input 2-differences, we performed several experiments with 100 random tests each. In each test, for every valid α_{0_S}, the number of iterations to enumerate 2-differences backwards is equal to the value computed based on Eq. 1 and the number of iterations to enumerate all compact 2-differential trails is much smaller than it. As for the guessing times to recover the master key, it is found that the obtained value indeed matches well with the theoretical value computed based on Eqs. 5 or 6.

8 Conclusion

In this paper, we present a 2-difference enumeration attack framework to analyze 4-round LowMC with a full S-box layer. With only 3 chosen plaintexts, we attack the 4-round LowMC instances adopting a full S-box layer with block size of 129, 192, and 255 bits, respectively. All these attacks have either a lower time complexity or a higher success probability than those proposed in the previous CRYPTO paper [19].

Acknowledgements We thank anonymous referees for their insightful comments. This work was supported by the National Key Research and Development Program of China (Grant Nos. 2022YFB2701900, 2018YFA0704704), the National Natural Science Foundation of China (Grant Nos. 62032014, 62202444), the Fundamental Research Funds for the Central Universities.

Code availability The code for our experiments is available at https://github.com/wxqiao/LowMC_new_attack_2diff.

Appendix A: A simplified proof for the generalization of observation 4

We denote the input and output of the 3-bit APN S-box by (x_0, x_1, x_2) and (y_0, y_1, y_2), respectively.

For a 3-bit APN S-box, its algebraic degree must be 2. Thus, it can be defined in the following way:

$$y_0 = \phi_0(x_0, x_1, x_2) \oplus a_0x_0x_1 \oplus a_1x_0x_2 \oplus a_2x_1x_2 \oplus c_0,$$
$$y_1 = \phi_1(x_0, x_1, x_2) \oplus a_3x_0x_1 \oplus a_4x_0x_2 \oplus a_5x_1x_2 \oplus c_1,$$
$$y_2 = \phi_2(x_0, x_1, x_2) \oplus a_6x_0x_1 \oplus a_7x_0x_2 \oplus a_8x_1x_2 \oplus c_2,$$

where $\phi_i(x_0, x_1, x_2)(0 \le i \le 2)$ are linear boolean functions and $a_j \in \mathrm{F}_2(0 \le j \le 8)$, $c_j \in \mathrm{F}_2(0 \le j \le 2)$.

Since Observation 2 holds for all 3-bit APN S-box [19], the generalization holds for the case when $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (\Delta x_0^2, \Delta x_1^2, \Delta x_2^2)$, or $(\Delta x_0^1, \Delta x_1^1, \Delta x_2^1) = (0, 0, 0)$, or $(\Delta x_0^2, \Delta x_1^2, \Delta x_2^2) = (0, 0, 0)$.

For the other case, we can write the accurate 8 valid output 2-differences, and it can be found that the 8 valid output 2-differences form an affine space of dimension 3.

For example, when input 2-difference is (0, 0, 1, 1, 0, 1), the possible output 2-differences are listed below:

$$\begin{aligned}
(x_0, x_1, x_2) \rightarrow & (\Delta y_0^1, \Delta y_1^1, \Delta y_2^1, \Delta y_0^2, \Delta y_1^2, \Delta y_2^2)\\
(0, 0, 0) \rightarrow & (\Delta\phi_0, \Delta\phi_1, \Delta\phi_2, \widetilde{\Delta\phi_0} \oplus a_1, \widetilde{\Delta\phi_1} \oplus a_4, \widetilde{\Delta\phi_2} \oplus a_7),\\
(0, 0, 1) \rightarrow & (\Delta\phi_0, \Delta\phi_1, \Delta\phi_2, \widetilde{\Delta\phi_0}, \widetilde{\Delta\phi_1}, \widetilde{\Delta\phi_2}),\\
(0, 1, 0) \rightarrow & (\Delta\phi_0 \oplus a_2, \Delta\phi_1 \oplus a_5, \Delta\phi_2 \oplus a_8, \widetilde{\Delta\phi_0} \oplus a_0 \oplus a_1 \oplus a_2, \widetilde{\Delta\phi_1} \oplus a_3 \oplus\\
& a_4 \oplus a_5, \widetilde{\Delta\phi_2} \oplus a_6 \oplus a_7 \oplus a_8),\\
(1, 0, 0) \rightarrow & (\Delta\phi_0 \oplus a_1, \Delta\phi_1 \oplus a_4, \Delta\phi_2 \oplus a_7, \widetilde{\Delta\phi_0}, \widetilde{\Delta\phi_1}, \widetilde{\Delta\phi_2}),\\
(0, 1, 1) \rightarrow & (\Delta\phi_0 \oplus a_2, \Delta\phi_1 \oplus a_5, \Delta\phi_2 \oplus a_8, \widetilde{\Delta\phi_0} \oplus a_0 \oplus a_2, \widetilde{\Delta\phi_1} \oplus a_3 \oplus a_5,\\
& \widetilde{\Delta\phi_2} \oplus a_6 \oplus a_8),\\
(1, 0, 1) \rightarrow & (\Delta\phi_0 \oplus a_1, \Delta\phi_1 \oplus a_4, \Delta\phi_2 \oplus a_7, \widetilde{\Delta\phi_0} \oplus a_1, \widetilde{\Delta\phi_1} \oplus a_4, \widetilde{\Delta\phi_2} \oplus a_7),\\
(1, 1, 0) \rightarrow & (\Delta\phi_0 \oplus a_1 \oplus a_2, \Delta\phi_1 \oplus a_4 \oplus a_5, \Delta\phi_2 \oplus a_7 \oplus a_8, \widetilde{\Delta\phi_0} \oplus a_0 \oplus a_2,\\
& \widetilde{\Delta\phi_1} \oplus a_3 \oplus a_5, \widetilde{\Delta\phi_2} \oplus a_6 \oplus a_8),\\
(1, 1, 1) \rightarrow & (\Delta\phi_0 \oplus a_1 \oplus a_2, \Delta\phi_1 \oplus a_4 \oplus a_5, \Delta\phi_2 \oplus a_7 \oplus a_8, \widetilde{\Delta\phi_0} \oplus a_0 \oplus a_1 \oplus\\
& a_2, \widetilde{\Delta\phi_1} \oplus a_3 \oplus a_4 \oplus a_5, \widetilde{\Delta\phi_2} \oplus a_6 \oplus a_7 \oplus a_8).
\end{aligned}$$

where we denote $\phi_i(0, 0, 1)$ by $\Delta\phi_i(0 \le i \le 2)$ and $\phi_i(1, 0, 1)$ by $\widetilde{\Delta\phi_i}(0 \le i \le 2)$.

As $\{(0, 0, 0, 0, 0, 0), (0, 0, 0, a_1, a_4, a_7), (a_2, a_5, a_8, a_0 \oplus a_1 \oplus a_2, a_3 \oplus a_4 \oplus a_5, a_6 \oplus a_7 \oplus a_8), (a_1, a_4, a_7, 0, 0, 0), (a_2, a_5, a_8, a_0 \oplus a_2, a_3 \oplus a_5, a_6 \oplus a_8), (a_1, a_4, a_7, a_1, a_4, a_7), (a_1 \oplus a_2, a_4 \oplus a_5, a_7 \oplus a_8, a_0 \oplus a_2, a_3 \oplus a_5, a_6 \oplus a_8), (a_1 \oplus a_2, a_4 \oplus a_5, a_7 \oplus a_8, a_0 \oplus a_1 \oplus a_2, a_3 \oplus a_4 \oplus a_5, a_6 \oplus a_7 \oplus a_8)\}$ is a linear subspace of dimension 3 of F_2^6, the 8 possible output 2-differences form an affine space of dimension 3.

Appendix B: A detailed explanation for the complexity T_3

In our attack using 3 chosen plaintexts, assume that there are t inactive S-boxes and j special-active S-boxes in the 3rd round. For each valid value of α_{0_S} with a inactive S-boxes and b special-active S-boxes in the 1st round, the number of values of α_2 such that the 2-difference transition $\alpha_{0\_S} \rightarrow \alpha_2$ is valid equals

$$M = 4^b \times 8^{m-a-b}.$$

When enumerating 2-differences backwards, the number of valid values of α_{2_S} which satisfies there are c inactive S-boxes and d special-active S-boxes in the 2nd round is

$$N_{cd}^0 = 4^j \times 8^{m-t-j} \times \binom{m}{c}\binom{m-c}{d}\left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d \times \left(\frac{42}{64}\right)^{m-c-d}.$$

And for each valid α_{2_S} which satisfies there are c inactive S-boxes and d special-active S-boxes in the 2nd round, the number of values of α_2 such that the 2-difference transition $\alpha_2 \leftarrow \alpha_{2\_S}$ is valid equals

$$N_{cd}^1 = 4^d \times 8^{m-c-d}.$$

Thus, the number of valid 4-round compact 2-differential trail where there are c inactive S-boxes and d special-active S-boxes in the 2nd round is

$$\begin{aligned} N_{cd} &= M \times N_{cd}^0 \times N_{cd}^1 \times 2^{-2n} \\ &= 4^b \times 8^{m-a-b} \times 4^j \times 8^{m-t-j} \times \binom{m}{c}\binom{m-c}{d}\left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d \\ &\quad \times \left(\frac{42}{64}\right)^{m-c-d} \times 4^d \times 8^{m-c-d} \times 2^{-2n}. \end{aligned}$$

For each valid 4-round compact 2-differential trail where there are c inactive S-boxes and d special-active S-boxes in the 2nd round, we need to iterate 2^{2c} times to guess 2 bits for each inactive S-box in the 2nd round. In each iteration, we construct $8m+3-3a-b-3t-j$ linear equations with $k+3t$ variables. If $k+3t \le 8m+3-3a-b-3t-j$, i.e. $6t+j \le 5m+3-3a-b$, then it can be expected that the equation system has at most 1 solution. The cost equals 1. If $k+3t > 8m+3-3a-b-3t-j$, i.e. $6t+j > 5m+3-3a-b$, then it can be expected that the equation system has $2^{6t+j-5m-3+3a+b}$ solutions. The cost equals $2^{6t+j-5m-3+3a+b}$.

Thus, if $k+3t \le 8m+3-3a-b-3t-j$, i.e. $6t+j \le 5m+3-3a-b$, the expected time complexity to retrieve the master key is

$$\begin{aligned} T_3 &= \sum_{c=0}^{m}\sum_{d=0}^{m-c} N_{cd} \times 2^{2c} \\ &= \sum_{c=0}^{m}\sum_{d=0}^{m-c} 4^b \times 8^{m-a-b} \times 4^j \times 8^{m-t-j} \times \binom{m}{c}\binom{m-c}{d}\left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d \\ &\quad \times \left(\frac{42}{64}\right)^{m-c-d} \times 4^d \times 8^{m-c-d} \times 2^{-2n} \times 2^{2c} \\ &\le 2^{2.73m-3t-j-3a-b}. \end{aligned} \tag{B1}$$

If $k+3t > 8m+3-3a-b-3t-j$, i.e. $6t+j > 5m+3-3a-b$, the expected time complexity to retrieve the master key is

$$T_3 = \sum_{c=0}^{m}\sum_{d=0}^{m-c} N_{cd} \times 2^{2c} \times 2^{6t+j-5m-3+3a+b}$$

$$= \sum_{c=0}^{m} \sum_{d=0}^{m-c} 4^b \times 8^{m-a-b} \times 4^j \times 8^{m-t-j} \times \binom{m}{c} \binom{m-c}{d} \left(\frac{1}{64}\right)^c \times \left(\frac{21}{64}\right)^d$$
$$\times \left(\frac{42}{64}\right)^{m-c-d} \times 4^d \times 8^{m-c-d} \times 2^{-2n} \times 2^{2c} \times 2^{6t+j-5m-3+3a+b}$$
$$\leq 2^{3t-2.27m-3}. \tag{B2}$$

References

1. Albrecht M.R., Rechberger C., Schneider T., Tiessen T., Zohner M.: Ciphers for MPC and FHE. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 430–454. Springer (2015)
2. Albrecht M., Grassi L., Rechberger C., Roy A., Tiessen T.: Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 191–219. Springer (2016)
3. Albrecht M.R., Grassi L., Perrin L., Ramacher S., Rechberger C., Rotaru D., Roy A., Schofnegger M.: Feistel structures for MPC, and more. In: European Symposium on Research in Computer Security. pp. 151–171. Springer (2019)
4. Banik S., Barooti K., Durak F.B., Vaudenay S.: Cryptanalysis of LowMC instances using single plaintext/ciphertext pair. IACR Trans. Symmetric Cryptol. pp. 130–146 (2020)
5. Banik S., Barooti K., Vaudenay S., Yan H.: New attacks on LowMC instances with a single plaintext/ciphertext pair. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 303–331. Springer (2021)
6. Banik S., Barooti K., Caforio A., Vaudenay S.: Memory-efficient single data-complexity attacks on LowMC using partial sets. Cryptology ePrint Archive (2022)
7. Brakerski Z., Gentry C., Vaikuntanathan V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory (TOCT) **6**(3), 1–36 (2014).
8. Canteaut A., Carpov S., Fontaine C., Lepoint T., Naya-Plasencia M., Paillier P., Sirdey R.: Stream ciphers: a practical solution for efficient homomorphic-ciphertext compression. J. Cryptol. **31**(3), 885–916 (2018).
9. Chase M., Derler D., Goldfeder S., Katz J., Kolesnikov V., Orlandi C., Ramacher S., Rechberger C., Slamanig D., Wang X., et al.: The picnic signature scheme (2020). https://microsoft.github.io/Picnic/
10. Dinur I.: Cryptanalytic applications of the polynomial method for solving multivariate equation systems over gf (2). In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 374–403. Springer (2021)
11. Dinur I., Liu Y., Meier W., Wang Q.: Optimized interpolation attacks on lowmc. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 535–560. Springer (2015)
12. Dobraunig C., Eichlseder M., Mendel F.: Higher-order cryptanalysis of lowmc. In: ICISC 2015. pp. 87–101. Springer (2015)
13. Dobraunig C., Eichlseder M., Grassi L., Lallemand V., Leander G., List E., Mendel F., Rechberger C.: Rasta: a cipher with low and depth and few ands per bit. In: Annual International Cryptology Conference. pp. 662–692. Springer (2018)
14. Dobraunig C., Grassi L., Guinet A., Kuijsters D.: Ciminion: symmetric encryption based on toffoli-gates over large finite fields. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–34. Springer (2021)
15. Gentry C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 169–178 (2009)
16. Goldreich O., Micali S., Wigderson A.: How to play any mental game, or a completeness theorem for protocols with honest majority. In: Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali, pp. 307–328 (2019)
17. Kales D., Zaverucha G.: Improving the performance of the picnic signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 154–188 (2020)
18. Kolesnikov V., Schneider T.: Improved garbled circuit: Free xor gates and applications. In: International Colloquium on Automata, Languages, and Programming. pp. 486–498. Springer (2008)
19. Liu F., Isobe T., Meier W.: Cryptanalysis of full lowmc and lowmc-m with algebraic techniques. In: Annual International Cryptology Conference. pp. 368–401. Springer (2021)

20. Liu F., Isobe T., Meier W.: Low-memory algebraic attacks on round-reduced LowMC. Cryptology ePrint Archive (2021)
21. Liu F., Meier W., Sarkar S., Isobe T.: New low-memory algebraic attacks on lowmc in the picnic setting. IACR Trans. Symmetr. Cryptol. pp. 102–122 (2022)
22. Liu F., Sarkar S., Wang G., Meier W., Isobe T.: Algebraic meet-in-the-middle attack on lowmc. Cryptology ePrint Archive (2022)
23. Méaux P., Journault A., Standaert F.X., Carlet C.: Towards stream ciphers for efficient FHE with low-noise ciphertexts. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 311–343. Springer (2016)
24. Nielsen J.B., Nordholt P.S., Orlandi C., Burra S.S.: A new approach to practical active-secure two-party computation. In: Annual Cryptology Conference. pp. 681–700. Springer (2012)
25. Nist's post-quantum cryptography competition, https://csrc.nist.gov/projects/post-quantum-cryptography
26. Peyrin T., Wang H.: The malicious framework: embedding backdoors into tweakable block ciphers. In: Annual International Cryptology Conference. pp. 249–278. Springer (2020)
27. Rechberger C., Soleimany H., Tiessen T.: Cryptanalysis of low-data instances of full lowmcv2. IACR Transactions on Symmetric Cryptology pp. 163–181 (2018)
28. Tiessen T.: Polytopic cryptanalysis. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 214–239. Springer (2016)
29. Yao A.C.: Protocols for secure computations. In: 23rd annual symposium on foundations of computer science (SFCS 1982). pp. 160–164. IEEE (1982)
30. Yao A.C.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (SFCS 1986). pp. 162–167. IEEE (1986)
31. Grassi L., Kales D., Rechberger C., Schofnegger M.: Survey of key-recovery attacks on lowmc in a single plaintext/ciphertext scenario (2020)