IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2024, No. 1, pp. 326–356. DOI:10.46586/tosc.v2024.i1.326-356

Finding Impossible Differentials in ARX Ciphers under Weak Keys

Qing Ling[1], Tingting Cui[1,2](✉), Hongtao Hu[1], Sijia Gong[1], Zijun He[1], Jiali Huang[1] and Jia Xiao[3]

[1] School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, China
[2] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Qingdao, China
[3] Huaxia Jingwei Information Technology Limited Company, Beijing, China
lingqing@hdu.edu.cn, cuitingting@hdu.edu.cn, huht@hdu.edu.cn, gongsijia@hdu.edu.cn, hezijun@hdu.edu.cn, huangjl@hdu.edu.cn, tonyalex2010@163.com

Abstract. Impossible differential cryptanalysis is very important in the field of symmetric ciphers. Currently, there are many automatic search approaches to find impossible differentials. However, these methods have two underlying assumptions: Markov cipher assumption and key independence assumption. Actually, these two assumptions are not true in ARX ciphers, especially lightweight ones. In this paper, we study the impossible differentials in ARX cipher under weak keys for the first time. Firstly, we propose several accurate difference propagation properties on consecutive two and three modular additions. Then, these properties are applied to four typical local constructions composed of two consecutive modular additions, two modular additions with a rotation operation, xoring secret key or constant in the middle, to find impossible differentials under weak keys or special constants. What's more, we propose a more accurate difference propagation property on three consecutive modular additions. It can be used to find impossible differentials on more complex local constructions under weak keys or special constants. In practical ciphers, these impossible differentials on local constructions can be used to find contradictions. Lastly, combining our new findings with traditional automatic search methods for impossible differentials, we propose a framework to find impossible differentials in ARX ciphers under weak keys. As applications, we apply the framework to SPECK-32/64, LEA and CHAM-64/128. As a result, we find two 8-round impossible differentials for SPECK-32/64 under 2^{60} weak keys, and one 11-round impossible differential for LEA under 2^{k-1} weak keys, where k is the key size. These impossible differentials can start from any round. Furthermore, we find two 22-round impossible differentials for CHAM-64/128 under 2^{127} weak keys starting from certain rounds. As far as we know, all these impossible differentials are longer than previous ones.

Keywords: Impossible differential · ARX cipher · Weak key

1 Introduction

1.1 Background

Impossible differential cryptanalysis (IDC) is one of the most powerful cryptanalysis methods in the field of symmetric ciphers. It was first introduced by Biham et al. and Knudsen to attack Skipjack [BBS99] and DEAL [Knu98], respectively. Unlike differential cryptanalysis (DC), IDC aims to find the longest (so-called best) differential characteristic with probability 0 instead of a differential characteristic with high probability. Using such an impossible differential (ID), all wrong keys would be filtered out in the key-recovery

Received: 2023-11-23 Accepted: 2024-01-23 Published: 2024-03-01

phase, while only the right key is remained. Since it was proposed, it has good performance on analyzing the security of many block ciphers, such as AES [MDRMH10], Camellia [Blo15], LEA [HLK+14], HIGHT [CWP12].

The key point of IDC for a target cipher is to find the best valid ID. At the very beginning, people found IDs by hand. They usually treated the underlying Substitution-box (S-box) used in the target cipher as the ideal S-box, i.e. all nonzero input and output difference transitions are possible, and relied on the linear layers to find IDs as long as possible. It is not always possible to find good IDs by hand because of the numerous possibilities of differential patterns passing complex linear layers. Later, several automatic search approaches were invented, such as $\mathcal{U}$-method [KHL10], $\mathcal{UID}$-method [LLWG14] as well as an extended tool by Wu and Wang in [WW12]. These methods also treat the S-boxes in target ciphers as ideal ones, but can use the linear layer more carefully to find IDs by computer. Over about twenty years, these methods are much popular to evaluate the resistance of target cipher against IDC.

However, S-boxes used in practical ciphers can never be ideal. Some input/output difference transitions under an S-box in reality will never happen. Taking the difference propagation through the S-box into consideration, it is possible to find longer IDs that could never be found with previous methods. Inspired by this, Sasaki et al. [ST17] and Cui et al. [CCJ+16] applied the MILP to impossible differential automatic search, respectively. Both described Differential Distribution Table (DDT) in their search models. As a result, they found longer IDs of LBlock, HIGHT, SHACAL-2, Midori128 and so on. Especially, Cui et al.'s automatic search method can be applied on ARX ciphers for the first time. Actually, the methods for searching differential characteristics based on SAT/SMT [AK18, KLT15, MP13, RKJ+20] or CP [SGL+17] can also be used to find ID by fixing its input and output differences.

As we summarize above, all methods to find ID are based on two underlying assumptions: Markov cipher assumption [LMM91] and key independence assumption. In the past decade, we have seen a trend in the cipher designs gearing towards lightweight applications. It usually implies a lighter round function, accompanied by a lighter key schedule for the cipher. This means that the validity of the Markov cipher assumption might be even less true for this new design paradigm due to less diffusion of round function, simpler key schedule, as well as more dependencies between internal state and secret round keys. Specifically, the fact is that most ARX ciphers are not true Markov ciphers due to round-key dependencies introduced by the key-schedule or consecutive modular additions without key injection. Therefore, searching for differential characteristics in ARX ciphers under the Markov cipher assumption may lead to incorrect probabilities. Ultimately, some differential characteristics searched out under Markov cipher assumption may never happen in practice.

Inspired by this, Beyne and Rijmen [BR22] studied differential cryptanalysis under fixed-key, developed a general methodology to analyze the fixed-key probabilities of differentials. They assert that several attacks are shown to be invalid, most others turn out to work only for some keys but can be improved for weak keys. Xu et al. [XLJ+22] proposed a SAT-based automatic search tool for impossible differential characteristics in ARX ciphers and found some distinguishers ignored by previous methods. Peyrin et al. [PT22] discussed a generic framework that cryptanalysts can use to analyze existing differential characteristics and provided a clear picture of the probability distribution for the cipher SKINNY as an example. However, some of these methods have basically verified their conclusions through a large number of experiments, and there is not enough theoretical analysis of the difference propagation patterns. Meanwhile, whether the differentials or impossible differentials do exist or not has remained unknown with the current theoretical understanding. The weak key space in the difference propagation of ciphers has not been determined.

In order to better evaluate the security of ARX ciphers, it is meaningful to evaluate the resistance against impossible differential cryptanalysis under weak keys. Inspired by that, we aim to propose a more accurate framework to find longer impossible differentials under weak keys and show how to find the weak key space theoretically.

1.2 Contribution

In this paper, we study the impossible differentials in ARX ciphers under weak keys. We propose more accurate XOR difference propagation properties on consecutive multiple modular additions according to ID patterns found experimentally. Then we apply them to some typical local constructions composed by two or three modular additions, some linear operations such as rotation, xor with key or constant under weak key. Based on our work and traditional automatic search method for ID, we propose a new framework to find IDs under weak keys. As applications, we use this framework on SPECK-32/64, LEA and CHAM-64/128, and find longer IDs. All the code can be found in our repository https://github.com/lingqing0707/ID-patterns. Our main contributions are listed as follows:

Propose more accurate XOR difference propagation properties on multiple modular additions. In [LM02], Lipmaa and Moriai studied the difference propagation patterns of modular addition and proposed two properties. Based on them, Fu et al. [FWG$^+$16] and Mouha et al. [MP13] proposed an automatic method to search differential characteristics in ARX ciphers based on MILP and SAT/SMT respectively. Recently, Li et al. [LGCX19] proposed a more accurate difference propagation property with constraints on input pair. In this paper, we further study the differential propagation patterns on consecutive additions. As a result, we find three impossible difference propagation properties for two consecutive modular additions. Within one of these properties, some internal state difference bits may need to be fixed. This may feel confusing because internal state differences can not be directly controlled. However, as the internal difference state of differential patterns passing consecutive several modular additions has been limited to a special set naturally in practical ciphers, which can meet the requirement on the internal state difference bits corresponding to that mentioned in the property.

Find three IDs for each of four local constructions and one ID for three more complex local constructions in ARX ciphers. By studying lots of ARX ciphers, we extract four typical local constructions, including two consecutive modular additions, two modular additions with a rotation, xor secret key or constant in the middle (please refer to Figure 4), from two consecutive round functions in ARX ciphers. Applying our new impossible difference propagation properties on these constructions, we obtain three IDs for each. Moreover, we extract three more complex local constructions about three consecutive modular additions from consecutive three round functions in ARX ciphers (please refer to Figure 5). For each construction, we find one ID. Note that some of these IDs are true only under weak keys. In practical ciphers, these IDs can be used in finding contradictions.

Propose a framework for finding IDs under weak keys. Based on traditional automatic search method to find differentials, truncated differentials and impossible differentials under key-independence assumption, as well as our new properties on local constructions, we propose a framework for finding IDs under weak keys. As far as we know, this is the first work to study ID in ARX ciphers under weak keys.

Applications. As applications, we apply the framework on SPECK cipher [BSS$^+$13] proposed by NSA, ISO standard cipher LEA [HLK$^+$14] and ultra lightweight cipher CHAM [KRK$^+$17]. As a result, we find two 8-round IDs for SPECK-32/64 under 2^{60} weak keys, one 11-round ID for LEA under 2^{k-1} weak keys, where k is the key size. All these IDs can start from any round. What's more, we find two 22-round IDs for CHAM-64/128 under 2^{127} weak keys starting from certain rounds. All IDs we found are longer than previous ones which are summarized in Table 1.

Table 1: Summary of impossible differential results for SPECK-32/64, LEA, CHAM-64/128

Cipher	Round	Weak key space	Starting round	Resource
SPECK-32/64	6	2^{64}	any	[LGCX18]
	6	2^{64}	any	[XSQ17]
	7	2^{64}	any	[LGCX19]
	8	$\mathbf{2^{60}}$	**any**	**Section 4.1**
LEA-k	10	2^{k}	any	[HLK+14]
	10	2^{k}	any	[CCJ+16]
	11	$\mathbf{2^{k-1}}$	**any**	**Section 4.2**
CHAM-64/128	18	2^{128}	any	[KRK+17]
	20	2^{128}	$i, i \in A$	[XLJ+22]
	22	$\mathbf{2^{127}}$	$\boldsymbol{i, i \in B}$	**Section 4.3**

[1] $A = \{3, 5, 11, 13, 19, 21, 27, 29, 35, 37, 43, 45, 51, 53, 59\}$.
[2] $B = \{2, 4, 10, 12, 18, 20, 26, 28, 34, 36, 42, 44, 50, 52, 58\}$.

1.3 Outline

This paper is organized as follows. Some preliminaries are given in Section 2. In Section 3, we propose more accurate XOR difference propagation properties on modular additions and a general framework for finding IDs under weak keys. Then, we apply this framework to SPECK-32/64, LEA and CHAM-64/128 in Section 4. In Section 5, we conclude this paper.

2 Preliminaries

2.1 Notations

General notations used in this paper are shown in the following list.

$x[i]$	The i-th bit of string x
$x[j : i]$	The i-th to the j-th bits of x, $i \leq j$
Δx	The XOR difference of two inputs x and x'
$x \boxplus y$	Addition of x and y modulo 2^n
$x \boxminus y$	Subtraction of x and y modulo 2^n
$x \oplus y$	Bitwise exclusive OR of x and y
$x \wedge y$	Bitwise AND of x and y
$\ggg \alpha$	Right circular shifts by α bits
$\lll \beta$	Left circular shifts by β bits
$\gg \alpha$	Right shifts by α bits
$\ll \beta$	Left shifts by β bits
$\neg x$	Bitwise NOT of x
$x \| y$	Concatenation of bit strings x and y
$(\overset{n-1}{*} \cdots \overset{i}{*} \cdots \overset{0}{*})$	n-bit string, the bit superscript indicates the position of the bit and $*$ is an undetermined bit whose value may be 0 or 1
$\Delta A \rightarrow \Delta B$	A possible differential pattern
$\Delta A \nrightarrow \Delta B$	An impossible differential pattern

2.2 Properties of Addition Modulo 2^n

Addition Modulo 2^n, denoted as $\boxplus$, is one of the important nonlinear components used in symmetric ciphers. We list some existing XOR difference propagation properties on modular addition.

Definition 1. (Addition modulo 2^n[LM02]). Let $z = x \boxplus y$, where $z, x, y \in \mathbb{F}_2^n$, then

$$\begin{cases} z[0] = x[0] \oplus y[0], \\ z[i] = x[i] \oplus y[i] \oplus c[i-1], \quad 1 \le i \le n-1. \end{cases}$$

where $c = (c[n-1], \cdots, c[1], c[0]) \in \mathbb{F}_2^n$ is the carry bit vector of $x \boxplus y$, defined recursively as:

$$\begin{cases} c[0] = x[0] \wedge y[0], \\ c[i] = (x[i] \wedge y[i]) \oplus (x[i] \wedge c[i-1]) \oplus (y[i] \wedge c[i-1]), \quad 1 \le i \le n-1. \end{cases} \tag{1}$$

In [LM02], Lipmaa and Moriai studied the XOR difference propagation patterns of modular addition. Their results are summarized in Property 1 and Property 2.

Property 1. [LM02] Let Δx, Δy and Δz be fixed n-bit XOR differences. The differential (Δx, $\Delta y \to \Delta z$) passing modular addition is possible if and only if

$$eq(\Delta x \lll 1, \Delta y \lll 1, \Delta z \lll 1) \wedge (\Delta x \oplus \Delta y \oplus \Delta z \oplus (\Delta y \lll 1)) = 0, \tag{2}$$

where

$$eq(x, y, z) = (\neg x \oplus y) \wedge (\neg x \oplus z). \tag{3}$$

Property 2. [LM02] Let Δx, Δy and Δz be fixed n-bit XOR differences. The differential (Δx, $\Delta y \to \Delta z$) passing modular addition is possible if and only if $\Delta x[0] \oplus \Delta y[0] \oplus \Delta z[0] = 0$ and $\Delta x[i-1] = \Delta y[i-1] = \Delta z[i-1] = \Delta x[i] \oplus \Delta y[i] \oplus \Delta z[i]$, for $\Delta x[i-1] = \Delta y[i-1] = \Delta z[i-1], 0 \le i \le n-1$.

These two properties are widely used in the automatic search of (impossible) differential characteristics based on MILP and SAT/SMT. Besides that, one more intuitive and one more accurate difference propagation properties of modular addition as Property 3 and Property 4 respectively have been proposed.

Property 3. [XSQ17] Let Δx, Δy and Δz be fixed n-bit XOR differences. Suppose that the differential (Δx, $\Delta y \to \Delta z$) passing modular addition is possible. If $l_1 = min\{i | \Delta x[i] = 1, 0 \le i \le n\}$, $l_2 = min\{i | \Delta y[i] = 1, 0 \le i \le n\}$ and $l = min\{l_1, l_2\}$, we have:

(1) If $l_1 = l_2 = l$, then $\Delta z[i] = 0$ for $0 \le i \le l$.

(2) If $l_1 \ne l_2$, then $\Delta z[l] = 1$, $\Delta z[i] = 0$ for $0 \le i \le l-1$.

Property 4. [LGCX19] Let Δx, Δy and Δz be fixed n-bit XOR differences, where $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$ and $\Delta z = z \oplus z'$. Suppose the differential (Δx, $\Delta y \to \Delta z$) passing modular addition is possible. If $\Delta x = \Delta y = (0 \cdots 0 \overset{l}{1} 0 \cdots 0)$, then $\Delta z = (0 \cdots 0)$ if and only if $x[l] \ne y[l]$ or $x'[l] \ne y'[l]$.

2.3 SPECK, LEA, and CHAM Block Ciphers

2.3.1 SPECK Block Cipher

SPECK [BSS+13] is a family of lightweight block ciphers published by the National Security Agency (NSA) in 2013. The instance with $2n$-bit block is denoted as SPECK-$2n$

where $n \in \{16, 24, 32, 48, 64\}$. SPECK-$2n$ with mn-bit key is referred as SPECK-$2n/mn$, $m \in \{2, 3, 4\}$. In total, SPECK has ten instances according to n and m. Each instance of SPECK uses the Feistel-like structure shown in Figure 1.

The key-dependent round function of SPECK-$2n$ can be written as:

$$R_k(x, y) = (((x \ggg \alpha) \boxplus y) \oplus k, (y \lll \beta) \oplus ((x \ggg \alpha) \boxplus y) \oplus k). \tag{4}$$

where the rotation amounts are $\alpha = 7$ and $\beta = 2$, if $n = 16$, as well as $\alpha = 8$ and $\beta = 2$, otherwise. Parameters for all instances of SPECK are specified in Table 2.

Table 2: All versions of SPECK family

Block size	Key size	n	m	α	β	Rounds
32	64	16	4	7	2	22
48	72	24	3	8	3	22
	96		4			23
64	96	32	3			26
	128		4			27
96	96	48	2			28
	144		3			29
128	128	64	2			32
	192		3			33
	256		4			34

Key Schedule. Let the mn-bit master key be $k = l_{m-2} \| l_{m-3} \| \cdots \| l_1 \| l_0 \| k_0$, where $k_0, l_i \in \mathbb{F}_2^n$, $0 \le i \le m - 2$. Sequences k_j and l_j are generated by

$$\begin{cases} l_{j+m-1} = (k_j + (l_j \ggg \alpha)) \oplus j, \\ k_{j+1} = (k_j \lll \beta) \oplus l_{j+m-1}. \end{cases}$$

Then, $k_j (j \ge 0)$ is the j-th round key. It is worth noting that m known consecutive round keys $k_j, \cdots, k_{j-m+1}$ are enough to derive the master key. For more information about SPECK, please refer to [BSS+13].

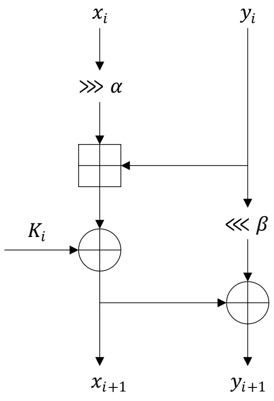


Figure 1: Round function of SPECK

2.3.2 LEA Block Cipher

LEA [HLK+14] is an ISO standard lightweight block cipher designed by Hong et al. It supports 128-bit block size and 128-bit, 192-bit, and 256-bit key sizes. The numbers of rounds for LEA are $r = 24$ for 128-bit key size, $r = 28$ for 192-bit key size and $r = 32$ for

256-bit key size. LEA's key-dependent round function (refer to Figure 2) can be written as:

$$
\begin{aligned}
x_{i+1}^0 &\leftarrow ((x_i^0 \oplus RK_i^0) \boxplus (x_i^1 \oplus RK_i^1)) \lll 9,\\
x_{i+1}^1 &\leftarrow ((x_i^1 \oplus RK_i^2) \boxplus (x_i^2 \oplus RK_i^3)) \ggg 5,\\
x_{i+1}^2 &\leftarrow ((x_i^2 \oplus RK_i^4) \boxplus (x_i^3 \oplus RK_i^5)) \ggg 3,\\
x_{i+1}^3 &\leftarrow x_i^0.
\end{aligned}
$$

where $x_i^0||x_i^1||x_i^2||x_i^3$ is the 128-bit input of the i-th round, $RK_i^0||RK_i^1||RK_i^2||RK_i^3||RK_i^4||RK_i^5$ is the 192-bit round key.

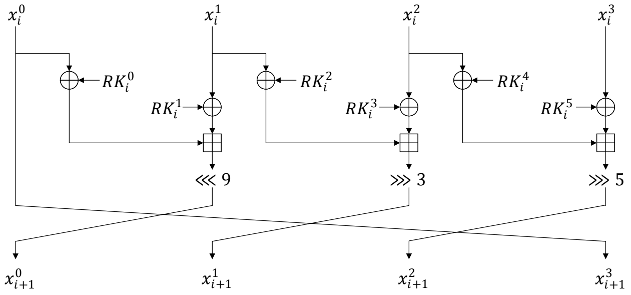


Figure 2: Round function of LEA

Key Schedule with a 128-bit key. Let $K = (K^0, K^1, K^2, K^3)$ be a 128-bit key. We set $T^j = K^j$ for $0 \leq j < 4$. Round key $RK_i, 0 \leq i < 24$ are generated as follows:

$$
\begin{aligned}
T^0 &\leftarrow (T^0 \boxplus (\delta^{i \bmod 4} \lll i)) \lll 1,\\
T^1 &\leftarrow (T^1 \boxplus (\delta^{i \bmod 4} \lll (i+1))) \lll 3,\\
T^2 &\leftarrow (T^2 \boxplus (\delta^{i \bmod 4} \lll (i+2))) \lll 6,\\
T^3 &\leftarrow (T^3 \boxplus (\delta^{i \bmod 4} \lll (i+3))) \lll 11,\\
RK_i &\leftarrow (T^0, T^1, T^2, T^1, T^3, T^1).
\end{aligned}
$$

where $\delta^0, \delta^1, \cdots, \delta^3$ are constants defined as:

$$\delta^0 = \texttt{c3efe9db}, \quad \delta^1 = \texttt{44626b02}, \quad \delta^2 = \texttt{79e27c8a}, \quad \delta^3 = \texttt{78df30ec}.$$

For more information about LEA, especially key schedules with 192-bit and 256-bit keys, please refer to [HLK+14].

2.3.3 CHAM Block Cipher

CHAM [KRK+17, RKJ+20] is a family of lightweight block ciphers based on ARX construction. It has three versions: CHAM-64/128, CHAM-128/128, and CHAM-128/256, where CHAM-n/k denotes the version with n-bit block and k-bit key. The total rounds of these three versions are 88, 112 and 120, respectively.

Divide the plaintext P into four w-bit words x_0^0, x_0^1, x_0^2 and x_0^3, where $P = x_0^0\|x_0^1\|x_0^2\|x_0^3$. Then the n-bit output X_{i+1} of the i-th round for $0 \leq i \leq r$ can be noted as $X_{i+1} = x_{i+1}^0\|x_{i+1}^1\|x_{i+1}^2\|x_{i+1}^3$. CHAM's key-dependent round function (depicted in Fig. 3) can be written as: if i is even,

$$
\begin{aligned}
x_{i+1}^3 &\leftarrow ((x_i^0 \oplus i) \boxplus ((x_i^1 \lll 1) \oplus (k_i \bmod 2k/w))) \lll 8,\\
x_{i+1}^j &\leftarrow x_i^{j+1}, \text{for } 0 \leq j \leq 2,
\end{aligned}
$$

otherwise,

$$\begin{aligned}x_{i+1}^3 &\leftarrow ((x_i^0 \oplus i) \boxplus ((x_i^1 \lll 8) \oplus (k_i \bmod 2k/w))) \lll 1,\\ x_{i+1}^j &\leftarrow x_i^{j+1}, \text{for } 0 \le j \le 2,\end{aligned}$$

where $k_i \bmod 2k/w$ is the i-th round key.

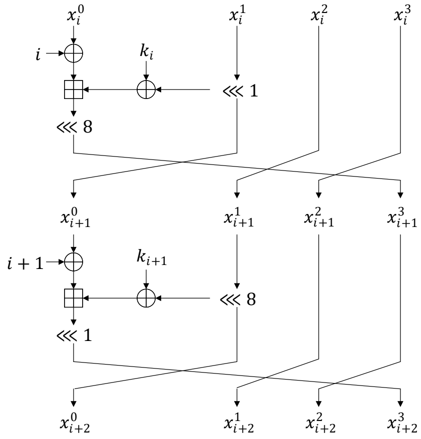


Figure 3: Two rounds of CHAM

Key Schedule. Let $K = K^0||K^1||\cdots||K^{k/w-1}$, then the $2k/w$ w-bit round keys $RK^0, RK^1, \cdots, RK^{2k/w-1}$ are generated as follows:

$$\begin{aligned}RK^j &\leftarrow K^j \oplus (K^j \lll 1) \oplus (K^j \lll 8),\\ RK^{(j+k/w)\oplus 1} &\leftarrow K^j \oplus (K^j \lll 1) \oplus (K^j \lll 11),\end{aligned}$$

where $0 \le j < k/w$ and $k_i \leftarrow RK^{i \bmod 2k/w}$.

3 Framework for Finding Impossible Differential Characteristics Under Weak Key in ARX Ciphers

In this section, we further study more accurate XOR difference propagation properties on multiple consecutive modular additions in Section 3.1. Based on these new properties, we propose a framework to find impossible differentials in ARX ciphers under weak keys in Section 3.2.

3.1 More Accurate XOR Difference Propagation Properties on Modular Addition

In this part, we first propose an accurate difference propagation property on Modular Subtraction shown in Property 5. Then we describe three ID properties on two consecutive modular additions depicted in Properties 6, 7, 9, as well as apply them on four typical local constructions extracted from two consecutive round functions of ARX ciphers as in Table 3∼6. What's more, we study the more accurate ID properties on three consecutive modular additions shown in Property 9, and apply it on three more complex local constructions, please refer to Table 7.

Property 5. Let $x = z \boxminus y$ and $x' = z' \boxminus y'$, where $x, y, z, x', y', z' \in \mathbb{F}_2^n$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$ and $\Delta z = z \oplus z'$. If $\Delta z = \Delta y = (\overset{n-1}{0} \cdots 0\,\overset{l}{1}\,0 \cdots \overset{0}{0})$, $0 \leq l < n-1$, then $\Delta x = (\overset{n-1}{0} \cdots \overset{0}{0})$ if and only if $z[l] = y[l]$ or $z'[l] = y'[l]$.

Proof. Let $c, c' \in \mathbb{F}_2^n$ be the carry bit vectors used to calculate z and z', and $\Delta c = c \oplus c'$. According to Definition 1, we have

$$z[i] = x[i] \oplus y[i] \oplus c[i-1], \quad z'[i] = x'[i] \oplus y'[i] \oplus c'[i-1]. \tag{5}$$

Thus, their XOR difference is

$$\Delta z[i] = \Delta x[i] \oplus \Delta y[i] \oplus \Delta c[i-1]. \tag{6}$$

Suppose that $\Delta x = (\overset{n-1}{0} \cdots \overset{0}{0})$. As $\Delta z = \Delta y = (\overset{n-1}{0} \cdots 0\,\overset{l}{1}\,0 \cdots \overset{0}{0})$, we have

$$\Delta c[i] = 0, 0 \leq i \leq n-2, \tag{7}$$

according to Equation (6).

By Definition 1 and Equation (7), as well as $\Delta x[l] = 0$, $\Delta y[l] = 1$, we have

$$(x[l] \wedge y[l]) \oplus (y[l] \wedge c[l-1]) \oplus (x'[l] \wedge y'[l]) \oplus (y'[l] \wedge c'[l-1]) = 0.$$

Then

$$\begin{cases} x[l] \oplus c[l-1] = 0, & \textit{if } y[l] = 1; \\ x'[l] \oplus c'[l-1] = 0, & \textit{if } y[l] = 0. \end{cases}$$

Thus, $z[l] = x[l] \oplus y[l] \oplus c[l-1] = y[l]$ or $z'[l] = x'[l] \oplus y'[l] \oplus c'[l-1] = y'[l]$.

Conversely, suppose that $z[l] = y[l]$ or $z'[l] = y'[l]$. By $\Delta z = \Delta y = (\overset{n-1}{0} \cdots 0\,\overset{l}{1}\,0 \cdots \overset{0}{0})$ and Property 3, we have

$$\Delta x[i] = 0, \text{ for } 0 \leq i \leq l. \tag{8}$$

Thus, according to Equation (6), we have $\Delta c[l-1] = 0$. Due to Definition 1 again, we can obtain

$$\Delta c[l] = \Delta(x[l] \wedge y[l]) \oplus \Delta(x[l] \wedge c[l-1]) \oplus \Delta(y[l] \wedge c[l-1]).$$

As $\Delta c[l-1] = \Delta x[l] = 0$ and $\Delta y[l] = 1$, we can deduce that

$$\Delta c[l] = \begin{cases} x[l] \oplus c[l-1], & \textit{if } y[l] = 1; \\ x'[l] \oplus c'[l-1], & \textit{if } y[l] = 0. \end{cases}$$

Since $\Delta z[l] = \Delta y[l] = 1$, we have $z[l] \oplus y[l] = z'[l] \oplus y'[l]$. Then, due to Equation (5) and the supposition $z[l] = y[l]$ or $z'[l] = y'[l]$, we have

$$\Delta c[l] = \begin{cases} x[l] \oplus c[l-1] = z[l] \oplus y[l] = 0, & \textit{if } y[l] = 1; \\ x'[l] \oplus c'[l-1] = z'[l] \oplus y'[l] = 0, & \textit{if } y[l] = 0. \end{cases}$$

Thus

$$\Delta x[l+1] = \Delta z[l+1] \oplus \Delta y[l+1] \oplus \Delta c[l] = 0. \tag{9}$$

By Property 1, it is clear that

$$\Delta x[i] = 0, \text{ for } l+2 \leq i \leq n-1. \tag{10}$$

Summing up the Equation (8), (9) and (10), we have $\Delta x = (\overset{n-1}{0} \cdots \overset{0}{0})$. □

Notably, Property 4 and Property 5 reveal two special cases of modular addition differential propagation on two consecutive bits. Surprisingly, these two properties are also found by Bao et al. in ASIACRYPT 2023. They listed the concrete bit-wise conditions for valid differential propagation through modular addition. For more details please refer to [BLYZ23].

Property 6. Let $z = x \boxplus y$, $z' = x' \boxplus y'$, $h = z \boxplus g$ and $h' = z' \boxplus g'$, where $x, y, z, g, h, x', y', z', g', h' \in \mathbb{F}_2^5$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$, $\Delta z = z \oplus z'$, $\Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. If $\Delta z[2:1] \neq 00$, then we have

$$(\Delta x = 1000*, \Delta y = 00***, \Delta g = 0000* \nrightarrow \Delta h = 00***). \tag{11}$$

Proof. `Reductio ad absurdum.` Suppose $(\Delta x = 1000*, \Delta y = 00***, \Delta g = 0000* \rightarrow \Delta h = 00***)$. Let c_1, c_1', c_2 and c_2' be the carry bit vectors of $x \boxplus y, x' \boxplus y', z \boxplus g$ and $z' \boxplus g'$ respectively, $\Delta c_1 = c_1 \oplus c_1'$ and $\Delta c_2 = c_2 \oplus c_2'$. By Property 1, we have

$$\begin{aligned} h = z \boxplus g, \Delta g = 0000*, \Delta h = 00*** &\Rightarrow \Delta z[4] = 0, \text{ if } \Delta z[3] = 0; \\ z = x \boxplus y, \Delta x = 1000*, \Delta y = 00*** &\Rightarrow \Delta z[4] = 1, \text{ if } \Delta z[3] = 0. \end{aligned}$$

Thus, $\Delta z[3] = 1$. Due to Definition 1,

$$\Delta z[i] = \Delta x[i] \oplus \Delta y[i] \oplus \Delta c_1[i-1], \tag{12}$$
$$\Delta h[i] = \Delta z[i] \oplus \Delta g[i] \oplus \Delta c_2[i-1]. \tag{13}$$

Therefore, by bringing the corresponding bit difference values into Equation (12) and Equation (13) respectively, we have $\Delta c_1[2] = 1$ and $\Delta c_2[2] = 1$.

According to the value of $\Delta z[2]$, there are two cases.

Case 1: Suppose $\Delta z[2] = 1$. Since $\Delta x[2] = 0$, then $\Delta y[2] = \Delta c_1[1] \oplus 1$ by Equation (12). Due to Definition 1 again,

$$\Delta c_1[i] = \Delta(x[i] \wedge y[i]) \oplus \Delta(x[i] \wedge c_1[i-1]) \oplus \Delta(y[i] \wedge c_1[i-1]). \tag{14}$$

Thus,

$$\Delta c_1[2] = \begin{cases} \Delta c_1[1] \wedge (x[2] \oplus y[2]), & \textit{if } \Delta y[2] = 0; \\ \Delta y[2] \wedge (x[2] \oplus c_1[1]), & \textit{if } \Delta y[2] = 1. \end{cases}$$

$$\Rightarrow \begin{cases} c_1[2] = c_1[1] \text{ and } x[2] \oplus y[2] = 1, & \textit{if } \Delta y[2] = 0; \\ c_1[2] = y[2] \text{ and } x[2] \oplus c_1[1] = 1, & \textit{if } \Delta y[2] = 1. \end{cases}$$

While,

$$z[2] = x[2] \oplus y[2] \oplus c_1[1] = \begin{cases} 1 \oplus c_1[1] = 1 \oplus c_1[2], & \textit{if } \Delta y[2] = 0; \\ 1 \oplus y[2] = 1 \oplus c_1[2], & \textit{if } \Delta y[2] = 1. \end{cases}$$

Therefore,

$$z[2] = 1 \oplus c_1[2]. \tag{15}$$

On the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, by Definition 1,

$$\Delta c_2[i] = \Delta(z[i] \wedge g[i]) \oplus \Delta(z[i] \wedge c_2[i-1]) \oplus \Delta(g[i] \wedge c_2[i-1]). \tag{16}$$

Since $\Delta z[2] = 1, \Delta c_2[2] = 1, \Delta g[2] = 0$,

$$\Delta c_2[2] = \begin{cases} \Delta(z[2] \wedge c_2[1]), & \textit{if } \Delta c_2[1] = 1; \\ \Delta z[2] \wedge (g[2] \oplus c_2[1]), & \textit{if } \Delta c_2[1] = 0. \end{cases}$$

$$\Rightarrow \begin{cases} c_2[1] = z[2] = c_2[2], & \textit{if } \Delta y[2] = 0; \\ g[2] \oplus c_2[1] = 1 \text{ and } c_2[2] = z[2], & \textit{if } \Delta y[2] = 1. \end{cases}$$

Above,

$$z[2] = c_2[2]. \tag{17}$$

Furthermore,

a) If $\Delta z[4] = 1$. Since $\Delta x[4] = 1, \Delta y[4] = 0, \Delta g[4] = 0$ and $\Delta h[4] = 0$, then $\Delta c_1[3] = 0$ and $\Delta c_2[3] = 1$ due to Equation (12) and (13). Since $\Delta c_1[2] = 1, \Delta x[3] = 0, \Delta y[3] = 0$, by Equation (14), we can obtain

$$\Delta c_1[3] = \Delta c_1[2] \wedge (x[3] \oplus y[3]) = 1 \Rightarrow x[3] \oplus y[3] = 0.$$

Then by Definition 1, and according to Equation (15),

$$z[3] = x[3] \oplus y[3] \oplus c_1[2] = c_1[2] = 1 \oplus z[2]. \tag{18}$$

Meanwhile, on the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, since $\Delta z[3] = 1, \Delta c_2[2] = 1, \Delta g[3] = 0$ and Equation (16), we have

$$\Delta c_2[3] = \Delta(z[3] \wedge c_2[2]) = 1 \Rightarrow z[3] = c_2[2].$$

Thus by Equation (17)

$$z[3] = z[2]. \tag{19}$$

Equation (18) and Equation (19) constitute a contradiction.

b) If $\Delta z[4] = 0$. Similar to the deduction of a) and Equations (15) and (17), we can obtain

$$z[3] = 1 \oplus c_1[2] = z[2] \tag{20}$$

and

$$z[3] = 1 \oplus c_2[2] = 1 \oplus z[2] \tag{21}$$

which constitute a contradiction.

Case 2: Suppose $\Delta z[2] = 0$. Since $\Delta x[2] = 0$, $\Delta y[2] = \Delta c_1[1]$ by Equation (12). Due to $\Delta c_1[2] = 1$ and Equation (14), we can obtain

$$\begin{aligned} &\Delta c_1[2] = \Delta(y[2] \wedge c_1[1]) = 1 \\ &\Rightarrow \Delta y[2] = \Delta c_1[1] = \Delta c_1[2] = 1 \text{ and } y[2] = c_1[1] = c_1[2]. \end{aligned} \tag{22}$$

Due to $\Delta c_2[2] = 1, \Delta g[2] = 0$ and Equation (16), we have

$$\Delta c_2[2] = \Delta c_2[1] \wedge (z[2] \oplus g[2]) = 1 \Rightarrow \Delta c_2[1] = 1 \text{ and } z[2] \oplus g[2] = 1.$$

By Definition 1,

$$\begin{aligned} c_2[2] &= (z[2] \wedge g[2]) \oplus (z[2] \wedge c_2[1]) \oplus (g[2] \wedge c_2[1]) \\ &= 0 \oplus (z[2] \oplus g[2]) \wedge c_2[1] = c_2[1]. \end{aligned} \tag{23}$$

According to $\Delta z[2:1] \neq 00$, we have $\Delta z[1] = 1$. Since $\Delta x[1] = 0, \Delta g[1] = 0$ combined with $\Delta c_1[1] = 1, \Delta c_2[1] = 1$, then we can deduce that

$$z[1] = 1 \oplus c_1[1],\ z[1] = c_2[1].$$

Due to Equations (22) and (23), we have

$$z[1] = 1 \oplus c_1[2],\ z[1] = c_2[2].$$

Also, according to the value of $\Delta z[4]$, we have the same result in **Case 1** that

$$\begin{cases} z[3] = c_1[2] \\ z[3] = c_2[2] \end{cases} \text{if } \Delta z[4] = 1; \begin{cases} z[3] = 1 \oplus c_1[2] \\ z[3] = 1 \oplus c_2[2] \end{cases} \text{if } \Delta z[4] = 0.$$

Then we have the contradiction about the relation between the values of $z[3]$ and $z[1]$.

To sum up, $(\Delta x = 1000*,\ \Delta y = 00***, \Delta g = 0000* \nrightarrow \Delta h = 00***)$. □

Similarly, we find the other property about two consecutive modular additions in $\mathbb{F}_2^5$ as follows.

Property 7. Let $z = x \boxplus y$, $z' = x' \boxplus y'$, $h = z \boxplus g$ and $h' = z' \boxplus g'$, where $x, y, z, g, h, x', y', z', g', h' \in \mathbb{F}_2^5$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$, $\Delta z = z \oplus z'$, $\Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. Then

$$(\Delta x = 0000*, \Delta y = 0010*, \Delta g = 0000* \nrightarrow \Delta h = 1010*). \tag{24}$$

Proof. See Appendix A. □

Property 8. Let $z = x \boxplus y$, $z' = x' \boxplus y'$, $h = z \boxplus g$ and $h' = z' \boxplus g'$, where $x, y, z, g, h, x', y', z', g', h' \in \mathbb{F}_2^4$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$, $\Delta z = z \oplus z'$, $\Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. Then

$$(\Delta x = 0000,\ \Delta y = 00**, \Delta g = 0000 \nrightarrow \Delta h = 100*) \tag{25}$$

Proof. See Appendix B. □

For Property 8, it can be noticed that the carry brought by the difference of the least significant bit does not affect the result of the differential propagation property. The ID can be extended as $(\Delta x = 0000*,\ \Delta y = 00***, \Delta g = 0000* \nrightarrow \Delta h = 100**)$. Therefore, Property 8 still shows a propagation property of two consecutive 5-bit modular differential patterns.

So far, we have put forward three differential propagation properties about two consecutive modular additions $z = x \boxplus y, h = z \boxplus g$ in $\mathbb{F}_2^5$, which are Properties 6, 7, and 8. It can be seen that the least significant bits of these IDs are undetermined, which means that the feasibility of these IDs is not affected by the values of less significant bits. What's more, the carries brought by lower bits can not make these impossible differential transitions viable. Thus, for x, y, z, g, h, if we add some uncertain bits at higher and lower positions, the IDs still hold. For example, if $\Delta z[i{+}2{:}i{+}1] \neq 00$, then

$$\begin{aligned}(\Delta x = *\cdots* \overset{i+4,\cdots,i}{\boxed{1000*}} *\cdots*, \Delta y = *\cdots* \overset{i+4,\cdots,i}{\boxed{00***}} *\cdots*, \Delta g = *\cdots* \overset{i+4,\cdots,i}{\boxed{0000*}} *\cdots* \\ \nrightarrow \Delta h = *\cdots* \overset{i+4,\cdots,i}{\boxed{00***}} *\cdots*)\end{aligned}$$

according to Property 6.

By analyzing lots of ARX ciphers, we find that the round function is usually simple enough that we can extract four typical local constructions from two consecutive rounds. The first local construction shown in Figure 4(a) is two consecutive modular addition operations. We can directly apply Properties 6, 7, 8 on this local construction and find out three IDs under some constraints which are described in Table 3.

The second local construction shown in Figure 4(b) is two modular addition operations with a left rotation in the middle. Please note the left rotation is a special permutation, we can still easily apply Properties 6, 7, 8 on this local construction and find out three IDs under some constraints which are described in Table 4.

The third local construction shown in Figure 4(c) adds one more XOR operation in the middle of two consecutive modular addition operations to mix the secret key. Due to the freedom of secret key, all IDs found in Table 5 become possible under certain key. However, we can remove such keys from the key space so as to avoid all possible difference propagation patterns. In other word, we can still apply Properties 6, 7, 8 on this local construction and find out three IDs under weak key setting (described in Table 5).

The fourth local construction shown in Figure 4(d) adds one more XOR operation in the middle of two consecutive modular addition operations to mix a known constant.

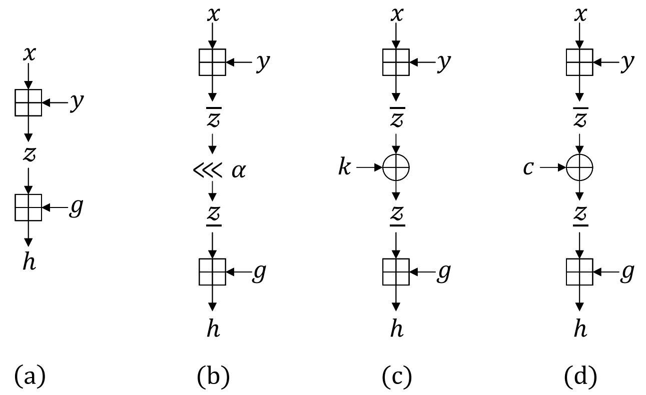


Figure 4: Four typical local constructions extracted from two consecutive rounds in ARX ciphers

Table 3: Three IDs on local construction shown in Figure 4(a)

Constraints	$\Delta z[i{+}2{:}i{+}1] \neq 00$		
Differentials	$\Delta x = (*\cdots* \overset{i+4,\cdots,i}{\boxed{1000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+3,\cdots,i}{\boxed{0000}} *\cdots*)$
	$\Delta y = (*\cdots* \overset{i+4,\cdots,i}{\boxed{00**}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0010*}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+3,\cdots,i}{\boxed{00**}} *\cdots*)$
	$\Delta z = (*\cdots* \overset{i+4,\cdots,i}{\boxed{*****}} *\cdots*)$	$\Delta z = (*\cdots* \overset{i+4,\cdots,i}{\boxed{*****}} *\cdots*)$	$\Delta z = (*\cdots* \overset{i+3,\cdots,i}{\boxed{****}} *\cdots*)$
	$\Delta g = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{i+3,\cdots,i}{\boxed{0000}} *\cdots*)$
	$\Delta h = (*\cdots* \overset{i+4,\cdots,i}{\boxed{00\,***}} *\cdots*)$	$\Delta h = (*\cdots* \overset{i+4,\cdots,i}{\boxed{1010*}} *\cdots*)$	$\Delta h = (*\cdots* \overset{i+3,\cdots,i}{\boxed{100*}} *\cdots*)$
Result	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 6	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 7	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 8

Table 4: Three IDs on local construction shown in Figure 4(b)

Constraints	$\Delta \overline{z}[i{+}2{:}i{+}1] \neq 00$		
Differentials	$\Delta x = (*\cdots* \overset{i+4,\cdots,i}{\boxed{1000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+3,\cdots,i}{\boxed{0000}} *\cdots*)$
	$\Delta y = (*\cdots* \overset{i+4,\cdots,i}{\boxed{00\,***}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+4,\cdots,i}{\boxed{0010*}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+3,\cdots,i}{\boxed{00**}} *\cdots*)$
	$\Delta \overline{z} = (*\cdots* \overset{i+4,\cdots,i}{\boxed{*****}} *\cdots*)$	$\Delta \overline{z} = (*\cdots* \overset{i+4,\cdots,i}{\boxed{*****}} *\cdots*)$	$\Delta \overline{z} = (*\cdots* \overset{i+3,\cdots,i}{\boxed{****}} *\cdots*)$
	$\Delta \underline{z} = (*\cdots* \overset{j+4,\cdots,j}{\boxed{*****}} *\cdots*)$	$\Delta \underline{z} = (*\cdots* \overset{j+4,\cdots,j}{\boxed{*****}} *\cdots*)$	$\Delta \underline{z} = (*\cdots* \overset{j+3,\cdots,j}{\boxed{****}} *\cdots*)$
	$\Delta g = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{j+3,\cdots,j}{\boxed{0000}} *\cdots*)$
	$\Delta h = (*\cdots* \overset{j+4,\cdots,j}{\boxed{00***}} *\cdots*)$	$\Delta h = (*\cdots* \overset{j+4,\cdots,j}{\boxed{1010*}} *\cdots*)$	$\Delta h = (*\cdots* \overset{j+3,\cdots,j}{\boxed{100*}} *\cdots*)$
Result	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 6	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 7	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 8

[1] The i-th bit of $\overline{z}$ is cyclically shifted to the j-th bit of $\underline{z}$.

Similar to the analysis that on the third local construction, we can find three IDs based on Properties 6, 7, 8 by putting some constraints on the constant (refer to Table 6). However the constant usually is the round constant that is fixed and known in advance. Thus, these IDs only apply to certain rounds in practical ciphers.

These impossible differentials were identified by experimentally obtaining the differential

Table 5: Three IDs on local construction shown in Figure 4(c)

Constraints	$k[i{+}3:i{+}1] = 000$ or 111 $\Delta\overline{z}[i{+}2:i{+}1] \neq 00$	$k[i{+}3:i{+}2] = 00$ or 11	$k[i{+}2:i{+}1] = 00$ or 11
Differentials	$\Delta x = (*\cdots*\overset{i+4,\cdots,i}{\boxed{1000*}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+4,\cdots,i}{\boxed{00***}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+4,\cdots,i}{\boxed{*****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+4,\cdots,i}{\boxed{00***}}*\cdots*)$	$\Delta x = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0010*}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+4,\cdots,i}{\boxed{*****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+4,\cdots,i}{\boxed{1010*}}*\cdots*)$	$\Delta x = (*\cdots*\overset{i+3,\cdots,i}{\boxed{0000}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+3,\cdots,i}{\boxed{00**}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+3,\cdots,i}{\boxed{****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+3,\cdots,i}{\boxed{0000}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+3,\cdots,i}{\boxed{100*}}*\cdots*)$
Result	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 6	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 7	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 8

Table 6: Three IDs on local construction shown in Figure 4(d)

Constraints	$c[i{+}3:i{+}1] = 000$ or 111 $\Delta\overline{z}[i{+}2:i{+}1] \neq 00$	$c[i{+}3:i{+}2] = 00$ or 11	$c[i{+}2:i{+}1] = 00$ or 11
Differentials	$\Delta x = (*\cdots*\overset{i+4,\cdots,i}{\boxed{1000*}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+4,\cdots,i}{\boxed{00***}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+4,\cdots,i}{\boxed{*****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+4,\cdots,i}{\boxed{00***}}*\cdots*)$	$\Delta x = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0010*}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+4,\cdots,i}{\boxed{*****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+4,\cdots,i}{\boxed{0000*}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+4,\cdots,i}{\boxed{1010*}}*\cdots*)$	$\Delta x = (*\cdots*\overset{i+3,\cdots,i}{\boxed{0000}}*\cdots*)$ $\Delta y = (*\cdots*\overset{i+3,\cdots,i}{\boxed{00**}}*\cdots*)$ $\Delta z = (*\cdots*\overset{i+3,\cdots,i}{\boxed{****}}*\cdots*)$ $\Delta g = (*\cdots*\overset{i+3,\cdots,i}{\boxed{0000}}*\cdots*)$ $\Delta h = (*\cdots*\overset{i+3,\cdots,i}{\boxed{100*}}*\cdots*)$
Result	$(\Delta x, \Delta y, \Delta g) \nrightarrow \Delta h)$ according to Property 6	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 7	$(\Delta x, \Delta y, \Delta g \nrightarrow \Delta h)$ according to Property 8

probability of all possible transitions over two consecutive five-bit modular additions. Actually, we obtained total hundreds of impossible differential patterns (under some classification of truncated differential) passing two continuous additions modulo 2^5, which are possible under the Markov hypothesis. It takes about 6 hours on a personal computer. Some of these ID patterns have been summarized in truncated impossible differentials shown in Properties 6∼8. There are other impossible differential patterns, such as the ID $(\Delta x = 0000*, \Delta y = 0000*, \Delta g = 0000* \nrightarrow \Delta h = 0100*)$ shown in our repository[1].

Next, we propose a more complex property on three consecutive modular additions which is described in Property 9. As in the practical ARX ciphers, rotation shift is often used in round function. Rotation shift moves bits in the higher position to the lower position. By Definition 1, the least significant bits of a modular addition triplet has special property: $z[0] = x[0] \oplus y[0]$, $c[0] = x[0] \wedge y[0]$ for $z = x \boxplus y$. Therefore, we should consider such property when analyzing differential propagation.

Property 9. $x \boxplus y = z (\mathrm{mod}\ 2^4)$, $x' \boxplus y' = z' (\mathrm{mod}\ 2^4)$, $z \boxplus d = e (\mathrm{mod}\ 2^4)$, $z' \boxplus d' = e' (\mathrm{mod}\ 2^4)$, $f \boxplus g = h (\mathrm{mod}\ 2^5)$ and $f' \boxplus g' = h' (\mathrm{mod}\ 2^5)$. Suppose that $\Delta x = x \oplus x', \Delta y = y \oplus y', \Delta z = z \oplus z', \Delta d = d \oplus d', \Delta e = e \oplus e', \Delta f = f \oplus f', \Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. If $f[4:1] = e$, then

$$(\Delta x = 000*, \Delta y = 000*, \Delta d = 10**, \Delta g = 0000* \nrightarrow \Delta h = 0000*) \tag{26}$$

Proof. See Appendix C. □

[1] https://github.com/lingqing0707/ID-patterns

The three modular additions in Property 9 can also be described as $\overline{z} = x \boxplus y, e = \underline{z} \boxplus d, h = f \boxplus g$, $\underline{z} = \overline{z}$ and f is still related to e as $f[4:1] = e$, where $x, y, \overline{z}, \underline{z}, d, e \in \mathbb{F}_2^4$ and $f, g, h \in \mathbb{F}_2^5$. Actually, if we add some uncertain bits at higher and lower positions for $x, y, \overline{z}, f, g, h$ and add some uncertain bits only at higher positions for $\underline{z}, d, e$, the ID $(\Delta x, \Delta y, \Delta d, \Delta g \nrightarrow \Delta h)$ in Property 9 still holds. The details are shown in Figure 5 and Table 7.

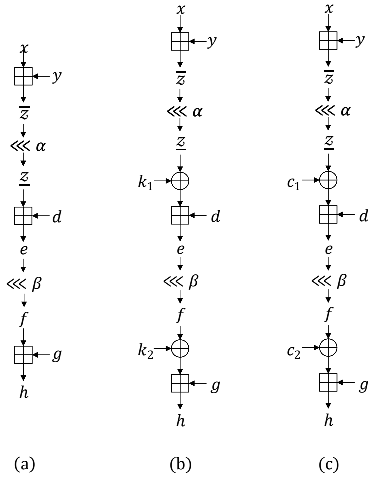


Figure 5: Three more complex constructions extracted from consecutive three round in ARX ciphers

Table 7: IDs on three more complex local constructions shown in Figure 5

Constraints		$k_1[2:1]$=00 or 11 $k_2[j+3:j+1]$ = 000 or 111	$c_1[2:1]$ = 00 or 11 $c_2[j+3:j+1]$ = 000 or 111
Differentials	$\Delta x = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$	$\Delta x = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$
	$\Delta y = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$	$\Delta y = (*\cdots* \overset{i+3,\cdots,i}{\boxed{000*}} *\cdots*)$
	$\Delta \overline{z} = (*\cdots* \overset{i+3,\ \cdots,\ i}{\boxed{****}} *\cdots*)$	$\Delta \overline{z} = (*\cdots* \overset{i+3,\ \cdots,\ i}{\boxed{****}} *\cdots*)$	$\Delta \overline{z} = (*\cdots* \overset{i+3,\ \cdots,\ i}{\boxed{****}} *\cdots*)$
	$\Delta \underline{z} = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$	$\Delta \underline{z} = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$	$\Delta \underline{z} = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$
	$\Delta d = (*\cdots* \overset{3,\cdots,0}{\boxed{10**}})$	$\Delta d = (*\cdots* \overset{3,\cdots,0}{\boxed{10**}})$	$\Delta d = (*\cdots* \overset{3,\cdots,0}{\boxed{10**}})$
	$\Delta e = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$	$\Delta e = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$	$\Delta e = (*\cdots* \overset{3,\ \cdots,\ 0}{\boxed{****}})$
	$\Delta f = (*\cdots* \overset{j+4,\ \cdots,\ j}{\boxed{*****}} *\cdots*)$	$\Delta f = (*\cdots* \overset{j+4,\ \cdots,\ j}{\boxed{*****}} *\cdots*)$	$\Delta f = (*\cdots* \overset{j+4,\ \cdots,\ j}{\boxed{*****}} *\cdots*)$
	$\Delta g = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta g = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$
	$\Delta h = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta h = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$	$\Delta h = (*\cdots* \overset{j+4,\cdots,j}{\boxed{0000*}} *\cdots*)$
Result	$(\Delta x, \Delta y, \Delta d, \Delta g \nrightarrow \Delta h)$ according to Property 9		

[1] The i-th bit of $\overline{z}$ is cyclically shifted to LSB of $\underline{z}$.

[2] The LSB of e is cyclically shifted to the j-th bit of f.

3.2 General Framework for Finding IDs Under Weak Key in ARX Ciphers

Until now, we can clearly find that ARX ciphers do not always follow the Markov cipher assumption. Based on the more accurate differential propagation properties in ARX ciphers, it is possible to find longer IDs for target cipher under weak keys, which may pose a threat to the security of ARX ciphers.

In this section, we propose a general framework to find IDs in ARX ciphers by combining our new findings with traditional automatic search method for ID based on STP. In our framework, different from almost impossible differential cryptanalysis [MDS10], we can deduce the exact weak key space for which the IDs hold. The framework is shown in Algorithm 1.

Algorithm 1 Finding IDs in ARX ciphers under weak keys

```
Initialization: Assign values to $i$ and $j$ so that $j-i$ is larger than the existing traditional ID rounds for the ARX cipher;
Determine the values of the $i$-th round input difference $\Delta x_i$ and the $j$-th round output difference $\Delta x_j$, which satisfy
    $$\Delta x_i \xrightarrow{Pro=1} \Delta x_{i+1} \text{ with some constraints on input } x_i, x'_i \text{ and round key } k_i$$
    $$\Delta x_{j-1} \xleftarrow{Pro=1} \Delta x_j \text{ with some constraints on output } x_j, x'_j \text{ and round key } k_j$$
    (via Property 1~Property 5);
Using Property 1, the values of $\Delta x_{i+m}$ and $\Delta x_{j-n}$ are obtained by means of automatic search tool for truncated differential and
    $$\Delta x_{i+1} \xrightarrow{Pro=1} \Delta x_{i+m}(\text{ or } \Delta x_i \xrightarrow{Pro=1} \Delta x_{i+m})$$
    $$\Delta x_{j-n} \xleftarrow{Pro=1} \Delta x_{j-1}(\text{ or } \Delta x_{j-n} \xleftarrow{Pro=1} \Delta x_j),$$
    where $i+m < j-n$;
According to Properties 6~9, check if there is contradiction from $\Delta x_{i+m}$ to $\Delta x_{j-n}$. Constraints on the internal state difference bits existed in some IDs can be checked by automatic search tool for finding differential characteristics.
If no contradiction, via automatic search tool for differential, find out the set of all possible cases of $\Delta x_{j+m+s}$. Then, split and classify the differential $\Delta x_{i+m} \rightarrow \Delta x_{j-n}$ into two parts as $\Delta x_{i+m} \rightarrow \Delta x_{j+m+s}$ and $\Delta x_{j+m+s} \rightarrow \Delta x_{j-n}$ according to the values of some bits of the internal state difference $\Delta x_{j+m+s}$.
Divide and conquer, for each part, use Properties 6~9 again to avoid the set by weak keys.
Output: $\Delta x_i \overset{(j-i)\text{-round}}{\nrightarrow} \Delta x_j$ and weak keys
    (with some constraints on input $x_i, x'_i$ and output $x_j, x'_j$).
```

4 Applications

We apply the new framework on practical ARX ciphers to find more accurate impossible differentials which were not found with traditional methods. As a result, we find two 8-round IDs for SPECK-32/64 under 2^{60} weak keys, one 11-round ID for LEA under 2^{k-1} weak keys, as well as two 22-round IDs for CHAM-64/128 under 2^{127} weak keys. All these IDs are longer than previous ones.

4.1 Applications to SPECK-32/64

According to the framework proposed in Section 3.2, we find two 8-round impossible differentials for SPECK-32/64 under 2^{60} weak keys.

The first one is that if $x_i[2] \neq y_i[11]$ (or $x'_i[2] \neq y'_i[11]$), $x_{i+8}[2] = x_{i+8}[4] \wedge y_{i+8}[4]$, we have

$$(\Delta x_i = 0 \cdots 0100, \Delta y_i = 000010 \cdots 0) \nrightarrow (\Delta x_{i+8} = 0 \cdots 010, \Delta y_{i+8} = 0 \cdots 01010)$$

under $k_{i+1}[14:13] = 00$ (or 11), $k_{i+3}[14:12] = 000$ (or 111), $k_{i+7}[1] = 0$.

The second one is that if $x_i[2] \neq y_i[11]$ or $(x'_i[2] \neq y'_i[11])$, $x_{i+8}[2] \neq x_{i+8}[4] \wedge y_{i+8}[4]$, we have

$$(\Delta x_i = 0 \cdots 0100, \Delta y_i = 000010 \cdots 0) \nrightarrow (\Delta x_{i+8} = 0 \cdots 010, \Delta y_{i+8} = 0 \cdots 01010)$$

under $k_{i+1}[14:13] = 00$ (or 11), $k_{i+3}[14:12] = 000$ (or 111), $k_{i+7}[1] = 1$.

Due to Property 4, if $x_i[2] \neq y_i[11]$ or $x'_i[2] \neq y'_i[11]$, at the first round, it can be deduced that

$$(\Delta x_i = 0 \cdots 0100, \Delta y_i = 000010 \cdots 0) \rightarrow (\Delta x_{i+1} = 0 \cdots 0, \Delta y_{i+1} = 0010 \cdots 0)$$

with Probability 1. Please refer to the red part at the first round in Figure 6.

Similarly, based on Property 5, if $k_{i+7}[1] = 0, x_{i+8}[1] = x_{i+8}[3] \wedge y_{i+8}[3]$ or $k_{i+7}[1] = 1, x_{i+8}[1] \neq x_{i+8}[3] \wedge y_{i+8}[3]$, at the last round, it can be deduced that

$$(\Delta x_{i+7} = 0 \cdots 0, \Delta y_{i+7} = 0 \cdots 010) \leftarrow (\Delta x_{i+8} = 0 \cdots 010, \Delta y_{i+8} = 0 \cdots 01010)$$

with Probability 1. Please refer to the red part at the last round in Figure 6.

Then, by traditional automatic search tool for impossible differential, we find that by analyzing the difference propagation, only two forms of $(\Delta x_{i+3}, \Delta y_{i+3})$ leads

$$(\Delta x_{i+1} = 0 \cdots 0, \Delta y_{i+1} = 0010 \cdots 0) \rightarrow (\Delta x_{i+7} = 0 \cdots 0, \Delta y_{i+7} = 0 \cdots 010)$$

to be possible. The two forms are

$$\Delta x_{i+3} = *******101000000, \Delta y_{i+3} = *********10000** \tag{27}$$

and

$$\Delta x_{i+3} = 10*******1000000), \Delta y_{i+3} = 00*******10000**. \tag{28}$$

This result can be obtained in few minutes. Meanwhile, $(\Delta x_{i+3}, \Delta y_{i+3})$ with form (28) results in that $\Delta x_{i+4}[13:12] \neq 00$.

As we would like to find ID from $(\Delta x_{i+1}, \Delta y_{i+1})$ to $(\Delta x_{i+7}, \Delta y_{i+7})$, we have to put some constraints on subkeys to avoid such two forms of $(\Delta x_{i+3}, \Delta y_{i+3})$.

Due to Table 5, we can get that if $k_{i+1}[14:13] = 00$ (or 11), $(\Delta x_{i+3}, \Delta y_{i+3})$ with form (27) will never happen. Please refer to the green part at rounds $i+1 \sim i+3$ in Figure 6.

Similarly, according to Table 5, we can obtain that if $k_{i+3}[14:12] = 000($ or $111)$, $(\Delta x_{i+3}, \Delta y_{i+3})$ with form (28) will never happen. Please refer to the blue part at round $i+3 \sim i+5$ in Figure 6.

In short, by putting constraints on $k_{i+1}[14:13], k_{i+3}[14:12]$ and $k_{i+7}[1]$ together, no differential of the form

$$(\Delta x_i = 0 \cdots 0100, \Delta y_i = 000010 \cdots 0) \rightarrow (\Delta x_{i+8} = 0 \cdots 010, \Delta y_{i+8} = 0 \cdots 01010)$$

will happen with probability larger than 0. In other words, we find two 8-round IDs for SPECK-32/64 under 2^{60} weak keys.

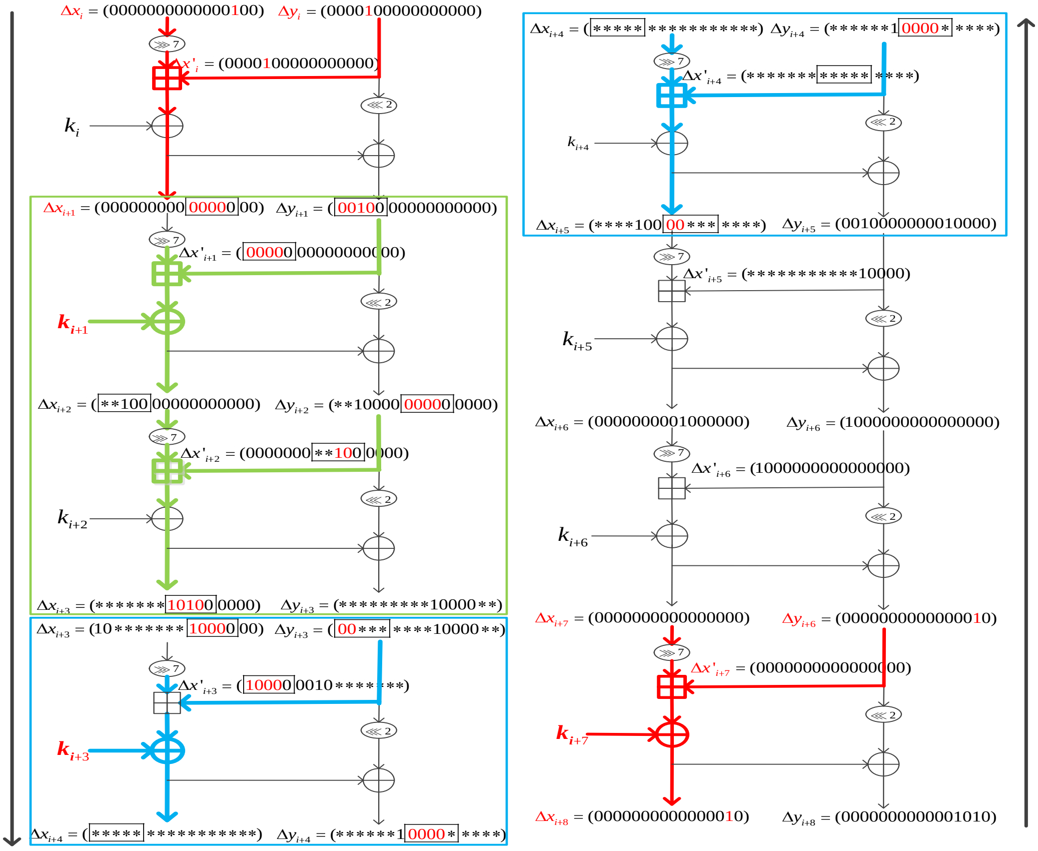

$\Delta x_{i} = (0000000000000100)$
$\Delta y_{i} = (0000100000000000)$
$\ggg 7$
$\Delta x'_{i} = (0000100000000000)$
$\lll 2$
k_{i}
$\Delta x_{i+1} = (000000000\boxed{00000}00)$
$\Delta y_{i+1} = (\boxed{00100}00000000000)$
$\Delta x'_{i+1} = (\boxed{00000}00000000000)$
k_{i+1}
$\Delta x_{i+2} = (\boxed{**100}00000000000)$
$\Delta y_{i+2} = (**10000\boxed{00000}0000)$
$\Delta x'_{i+2} = (0000000\boxed{**100}0000)$
k_{i+2}
$\Delta x_{i+3} = (*******\boxed{10100}0000)$
$\Delta y_{i+3} = (*********10000**)$
$\Delta x_{i+3} = (10*******\boxed{10000}00)$
$\Delta y_{i+3} = (\boxed{00***}****10000**)$
$\Delta x'_{i+3} = (\boxed{10000}0010*******)$
k_{i+3}
$\Delta x_{i+4} = (\boxed{*****}***********)$
$\Delta y_{i+4} = (******1\boxed{0000*}****)$
$\Delta x_{i+4} = (\boxed{*****}***********)$
$\Delta y_{i+4} = (******1\boxed{0000*}****)$
$\Delta x'_{i+4} = (*******\boxed{*****}****)$
k_{i+4}
$\Delta x_{i+5} = (****100\boxed{00***}****)$
$\Delta y_{i+5} = (0010000000010000)$
$\Delta x'_{i+5} = (***********10000)$
k_{i+5}
$\Delta x_{i+6} = (0000000001000000)$
$\Delta y_{i+6} = (1000000000000000)$
$\Delta x'_{i+6} = (1000000000000000)$
k_{i+6}
$\Delta x_{i+7} = (0000000000000000)$
$\Delta y_{i+6} = (0000000000000010)$
$\Delta x'_{i+7} = (0000000000000000)$
k_{i+7}
$\Delta x_{i+8} = (0000000000000010)$
$\Delta y_{i+8} = (0000000000001010)$

Figure 6: 8-round ID for SPECK-32/64

4.2 Applications to LEA

Here, we propose an 11-round impossible differential for LEA-128 under 2^{127} weak keys. That is

$$\begin{aligned}
&(\Delta x_i^0 = 10\cdots 0, \Delta x_i^1 = 10\cdots 0, \Delta x_i^2 = 10\cdots 0, \Delta x_i^3 = 10\cdots 0)\\
\nrightarrow\ &(\Delta x_{i+11}^0 = 0\cdots 0, \Delta x_{i+11}^1 = 00010\cdots 0, \Delta x_{i+11}^2 = 10\cdots 0, \Delta x_{i+11}^3 = 0\cdots 0)
\end{aligned}$$

under two fixed subkey bits $T_{i+6}^1[6:5] = 00$ or 11.

This ID can be split into three parts: rounds $i \sim i+5$, rounds $i+5 \sim i+7$ and the last rounds $i+7 \sim i+11$. By traditional IDC, we can easily get the input difference of round $i+6$ as

$$\begin{aligned}
&\Delta x_{i+5}^0 = *\cdots*\overset{9}{1}*\cdots*,\ \Delta x_{i+5}^1 = 0\cdots\overset{27}{0}*\cdots*\overset{13}{1}0\cdots 0,\\
&\Delta x_{i+5}^2 = 000*\cdots*\overset{19}{1}0\cdots 0,\ \Delta x_{i+5}^3 = 0\cdots\overset{9}{0}*\cdots*
\end{aligned} \tag{29}$$

and output difference of round $i+7$ as

$$\Delta x_{i+7}^0 = *\cdots*100),\ \Delta x_{i+7}^1 = *\cdots*100,\ \Delta x_{i+7}^2 = *\cdots*,\ \Delta x_{i+7}^3 = *\cdots* \tag{30}$$

depicted in Figure 9 and 10 respectively.

According to Table 5, if subkey bits $T_{i+6}^1[6:5] = 00$ (or 11), the difference with form (29) can not lead to the difference with form (30). Please refer to the red part in Figure 7.

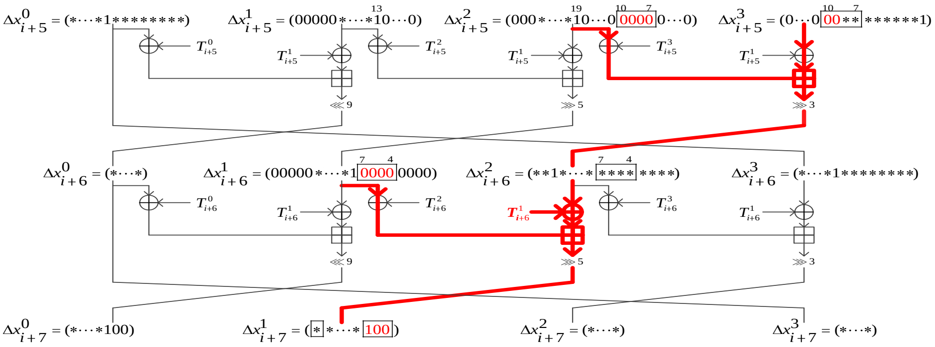

Δx^0_{i+5} = (∗···∗1∗∗∗∗∗∗∗∗)
Δx^1_{i+5} = (00000∗···∗10···0)
13
Δx^2_{i+5} = (000∗···∗10···0 0000 0···0)
19
10 7
Δx^3_{i+5} = (0···0 00∗∗ ∗∗∗∗∗∗1)
10 7
T^0_{i+5}
T^1_{i+5}
T^2_{i+5}
T^3_{i+5}
⋘9
⋙5
⋙3
Δx^0_{i+6} = (∗···∗)
Δx^1_{i+6} = (00000∗···∗1 0000 0000)
7 4
Δx^2_{i+6} = (∗∗1∗···∗ ∗∗∗∗ ∗∗∗∗)
7 4
Δx^3_{i+6} = (∗···∗1∗∗∗∗∗∗∗∗)
T^0_{i+6}
T^1_{i+6}
T^2_{i+6}
T^3_{i+6}
⋘9
⋙5
⋙3
Δx^0_{i+7} = (∗···∗100)
Δx^1_{i+7} = (∗ ∗···∗ 100)
Δx^2_{i+7} = (∗···∗)
Δx^3_{i+7} = (∗···∗)

Figure 7: Contradiction on the rounds $i + 5 \sim i + 6$ of 11-round ID for LEA-128

It is worth noting that all versions in the LEA family have such 11-round ID under 2^{k-1} weak keys, where k is the size of key. This fact can be found simply by comparing the key schedule between different versions. Meanwhile, such ID can start from any round.

Table 8: Comparison of IDs for LEA

ID	Rounds	Weak key space	Resource
$(1^1 0^{31}, 1^1 0^{31}, 1^1 0^{31}, 1^1 0^{31}) \nrightarrow (0^{32}, 0^{32}, 0^3 1^1 0^{28}, 0^{32})$	10	2^k	[HLK+14]
$(1^1 0^{31}, 1^1 0^{31}, 1^1 0^{31}, 1^1 0^{31}) \nrightarrow (0^{32}, 0^{32}, 0^3 1^1 0^{28}, 0^{32})$	11	2^{k-1}	this paper

[1] 0^i(resp. 1^i) represents i consecutive bits of 0 (resp. 1).
[2] k is the size of key for LEA.

Discussion about key-recovery attack on LEA. We compare the new 11-round ID with the previous best 10-round ID [HLK+14] in Table 8. It is easy to find that both IDs have the same input/output difference, but 11-round ID is only valid under 2^{k-1} weak keys and has a more complex contradiction in the middle. Furthermore, both IDs can be truncated to truncated ID distinguishers as

$$(0^{32}, 0^{32}, 0^{32}, *^6 0^{26}) \nrightarrow (0^{32}, 0^{32}, 0^3 1^1 0^{28}, 0^{32}),$$

which can be used to implement the key-recovery attack.

Note that the truncated ID transformed from 11-round ID has 9 rounds, still one more round than that from 10-round ID. In the other hand, due to the key schedule of LEA, as long as the forms of input/output difference of distinguishers are the same, then their key-recovery phases are the same. Therefore, one can exploit the 11-round impossible differential to make a better key-recovery attack under 2^{k-1} weak keys than that mentioned in [HLK+14] by using a set of specially chosen plaintexts.

4.3 Applications to CHAM-64/128

By applying our method on CHAM-64/128, we find two 22-round impossible differentials for CHAM-64/128 under 2^{127} weak keys.

The first one is, if $x_i^0[7] \neq x_i^1[15]$, then

$$\begin{aligned}&(\Delta x_i^0 = 0 \cdots 0 \overset{7}{1} 0 \cdots 0, \Delta x_i^1 = 10 \cdots 0, \Delta x_i^2 = 0 \cdots 0, \Delta x_i^3 = 0 \cdots 0)\\ \nrightarrow\ &(\Delta x_{i+22}^0 = 01 \cdots 0, \Delta x_{i+22}^1 = 0 \cdots 0, \Delta x_{i+22}^2 = 0 \cdots 0, \Delta x_{i+22}^3 = 0 \cdots 0 \overset{7}{1} 0)\end{aligned}$$

under $k_i[7] = 0$.

The second one is, if $x_i^0[7] = x_i^1[15]$, then

$$\begin{aligned}&(\Delta x_i^0 = 0\cdots 0\overset{7}{1}0\cdots 0, \Delta x_i^1 = 10\cdots 0, \Delta x_i^2 = 0\cdots 0, \Delta x_i^3 = 0\cdots 0)\\ \nrightarrow\ &(\Delta x_{i+22}^0 = 01\cdots 0, \Delta x_{i+22}^1 = 0\cdots 0, \Delta x_{i+22}^2 = 0\cdots 0, \Delta x_{i+22}^3 = 0\cdots 0\overset{7}{1}0)\end{aligned}$$

under $k_i[7] = 1$.

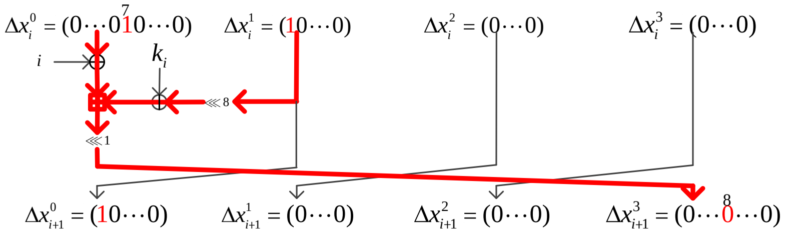
These IDs are only established when starting from the i-th round, where

$$i \in \{2, 4, 10, 12, 18, 20, 26, 28, 34, 36, 42, 44, 50, 52, 58\}.$$

According to Property 4, if $K_i[7] = 0, x_i^0[7] \neq x_i^1[15]$ or $K_i[7] = 1, x_i^0[7] = x_i^1[15]$, after the first round, the output difference will be

$$(\Delta x_{i+1}^0 = 10\cdots 0, \Delta x_{i+1}^1 = 0\cdots 0, \Delta x_{i+1}^2 = 0\cdots 0, \Delta x_{i+1}^3 = 0\cdots 0)$$

with Probability 1. Please refer to the red part in Figure 8. Then, by traditional IDC, we can obtain the output difference of round $i + 8$ as



$\Delta x_i^0 = (0\cdots 0\overset{7}{1}0\cdots 0)$
$\Delta x_i^1 = (10\cdots 0)$
$\Delta x_i^2 = (0\cdots 0)$
$\Delta x_i^3 = (0\cdots 0)$
i
k_i
$\lll 8$
$\lll 1$
$\Delta x_{i+1}^0 = (10\cdots 0)$
$\Delta x_{i+1}^1 = (0\cdots 0)$
$\Delta x_{i+1}^2 = (0\cdots 0)$
$\Delta x_{i+1}^3 = (0\cdots \overset{8}{0}\cdots 0)$

Figure 8: Contradiction on the i-th round of 22-round ID for CHAM-64/128

$$(\Delta x_{i+9}^0 = 10000000*\cdots*, \Delta x_{i+9}^1 = 0\cdots 0, \Delta x_{i+9}^2 = ******10*\cdots*, \Delta x_{i+9}^3 = *\cdots*1*)$$

and the input difference of round $i + 18$ as

$$(\Delta x_{i+18}^0 = **10\cdots 0, \Delta x_{i+18}^1 = 0\cdots 0, \Delta x_{i+18}^2 = 0\cdots 0, \Delta x_{i+18}^3 = 0\cdots 0),$$

which are shown in Table 9 and 10 respectively.

Within the middle remained 9 rounds, according to Table 7 in Appendix E, we have

$$\begin{aligned}&(\Delta x_{i+9}^0 = 10000000*\cdots*, \Delta x_{i+9}^1 = 0\cdots 0, \Delta x_{i+9}^2 = ******10*\cdots*, \Delta x_{i+9}^3 = *\cdots*1*)\\ \nrightarrow\ &(\Delta x_{i+18}^0 = **10\cdots 0, \Delta x_{i+18}^1 = 0\cdots 0, \Delta x_{i+18}^2 = 0\cdots 0, \Delta x_{i+18}^3 = 0\cdots 0)\end{aligned}$$

under constants bits $(i + 13)[2 : 1] = 00$ or 11 and $(i + 17)[10 : 8] = 000$ or 111, which is depicted in the red part of Figure 11 in Appendix F.

Please note that the conditions are on the known constants. The IDs above must start from specific rounds, such as the $\{2, 4, 10, 12, 18, 20, 26, 28, 34, 36, 42, 44, 50, 52, 58\}$ -th round.

In short, we find two 22-round impossible differentials for CHAM-64/128 under 2^{127} weak keys.

5 Conclusion

In this paper, we propose a framework to find impossible differentials of ARX ciphers under weak key for the first time. As applications, we consider SPECK, LEA and CHAM to find longer IDs under weak key. Actually, there is much further work along this direction. As properties 6 to 8 and 9 represent just a thin selection of the impossible differentials found experimentally, it is valuable to continue analyzing these ID patterns. It is also a meaningful work to try to build an automated search model to find more impossible differentials like the ones used in this paper. Trying to build a new automatic search model for impossible differentials of target ARX ciphers under weak keys is deserving as well. On the other hand, we evaluated the effectiveness of these new impossible differentials in key recovery attacks and found that some of them are indeed not good, but some could be better than previous ones, such as the distinguisher we proposed for LEA. In fact, there are still many differential patterns that we have not used yet. It is conceivable that some patterns could be used to further extend the impossible differential distinguishers of ARX ciphers to solve the problem in key recovery attacks. Therefore, it is worthwhile to dig deeper for more impossible differentials to get better key recovery attacks for ARX ciphers.

Acknowledgments

This work is supported by the Major Basic Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025).

References

[AK18] Ralph Ankele and Stefan Kölbl. Mind the gap-a closer look at the security of block ciphers against differential cryptanalysis. In *International Conference on Selected Areas in Cryptography*, pages 163–190. Springer, 2018.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In *Advances in Cryptology–EUROCRYPT 1999: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*, pages 12–23. Springer, 1999.

[Blo15] Céline Blondeau. Impossible differential attack on 13-round camellia-192. *Information Processing Letters*, 115(9):660–666, 2015.

[BLYZ23] Zhenzhen Bao, Jinyu Lu, Yiran Yao, and Liu Zhang. More insight on deep learning-aided cryptanalysis. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 436–467. Springer, 2023.

[BR22] Tim Beyne and Vincent Rijmen. Differential cryptanalysis in the fixed-key model. In *Annual International Cryptology Conference*, pages 687–716. Springer, 2022.

[BSS$^+$13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *cryptology eprint archive*, 2013.

[CCJ$^+$16] Tingting Cui, Shiyao Chen, Keting Jia, Kai Fu, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. *Cryptology ePrint Archive*, 2016.

[CWP12] Jiazhe Chen, Meiqin Wang, and Bart Preneel. Impossible differential cryptanalysis of the lightweight block ciphers tea, xtea and hight. In *Progress in Cryptology–AFRICACRYPT 2012: 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings 5*, pages 117–137. Springer, 2012.

[FWG$^+$16] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In *Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23*, pages 268–288. Springer, 2016.

[HLK$^+$14] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. Lea: A 128-bit block cipher for fast encryption on common processors. In *Information Security Applications: 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers 14*, pages 3–27. Springer, 2014.

[KHL10] Jongsung Kim, Seokhie Hong, and Jongin Lim. Impossible differential cryptanalysis using matrix method. *Discrete Mathematics*, 310(5):988–1002, 2010.

[KLT15] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the simon block cipher family. In *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I 35*, pages 161–185. Springer, 2015.

[Knu98] Lars Knudsen. Deal-a 128-bit block cipher. *complexity*, 258(2):216, 1998.

[KRK$^+$17] Bonwook Koo, Dongyoung Roh, Hyeonjin Kim, Younghoon Jung, Dong-Geon Lee, and Daesung Kwon. Cham: A family of lightweight block ciphers for resource-constrained devices. In *International conference on information security and cryptology*, pages 3–25. Springer, 2017.

[LGCX18] Mingming Li, Jiansheng Guo, Jingyi Cui, and Linhong Xu. Analysis of impossible differential characteristics for speck families of block ciphers. *Journal of Cryptologic Research*, 2018.

[LGCX19] Mingming Li, Jiansheng Guo, Jingyi Cui, and Linhong Xu. Impossible differential cryptanalysis of speck. In *Trusted Computing and Information Security: 12th Chinese Conference, CTCIS 2018, Wuhan, China, October 18, 2018, Revised Selected Papers 12*, pages 16–31. Springer, 2019.

[LLWG14] Yiyuan Luo, Xuejia Lai, Zhongming Wu, and Guang Gong. A unified method for finding impossible differentials of block cipher structures. *Information Sciences*, 263:211–220, 2014.

[LM02] Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In *Fast Software Encryption: 8th International Workshop, FSE 2001 Yokohama, Japan, April 2–4, 2001 Revised Papers 8*, pages 336–350. Springer, 2002.

[LMM91] Xuejia Lai, James L Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*, pages 17–38. Springer, 1991.

[MDRMH10] Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round aes-128. In *Progress in Cryptology–INDOCRYPT 2010: 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings 11*, pages 282–291. Springer, 2010.

[MDS10] Hamid Mala, Mohammad Dakhilalian, and Mohsen Shakiba. Cryptanalysis of block ciphers using almost-impossible differentials. *Cryptology ePrint Archive*, 2010.

[MP13] Nicky Mouha and Bart Preneel. Towards finding optimal differential characteristics for arx: Application to salsa20. *Cryptology ePrint Archive*, 2013.

[PT22] Thomas Peyrin and Quan Quan Tan. Mind your path: On (key) dependencies in differential characteristics. *Cryptology ePrint Archive*, 2022.

[RKJ+20] Dongyoung Roh, Bonwook Koo, Younghoon Jung, Il Woong Jeong, Dong-Geon Lee, Daesung Kwon, and Woo-Hwan Kim. Revised version of block cipher cham. In *Information Security and Cryptology–ICISC 2019: 22nd International Conference, Seoul, South Korea, December 4–6, 2019, Revised Selected Papers 22*, pages 1–19. Springer, 2020.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of aes, skinny, and others with constraint programming. *IACR transactions on symmetric cryptology*, 2017(1):281–306, 2017.

[ST17] Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects: Revealing structural properties of several ciphers. In *Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part III 36*, pages 185–215. Springer, 2017.

[WW12] Shengbao Wu and Mingsheng Wang. Automatic search of truncated impossible differentials for word-oriented block ciphers. In *International Conference on Cryptology in India*, pages 283–302. Springer, 2012.

[XLJ+22] Zheng Xu, Yongqiang Li, Lin Jiao, Mingsheng Wang, and Willi Meier. Do not misuse the markov cipher assumption-automatic search for differential and impossible differential characteristics in arx ciphers. *Cryptology ePrint Archive*, 2022.

[XSQ17] Hong XU, Penghui SU, and Wenfeng QI. Impossible differential cryptanalysis of reduced-round speck. *Journal of Electronics & Information Technology*, 39(10):2479–2486, 2017.

A The proof of Property 7

Property 7. Let $z = x \boxplus y$, $z' = x' \boxplus y'$, $h = z \boxplus g$ and $h' = z' \boxplus g'$, where $x, y, z, g, h, x', y', z', g', h' \in \mathbb{F}_2^5$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$, $\Delta z = z \oplus z'$, $\Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. Then

$$(\Delta x = 0000{*}, \Delta y = 0010{*}, \Delta g = 0000{*} \nrightarrow \Delta h = 1010{*}). \tag{31}$$

Proof. `Reductio ad absurdum.` Suppose $(\Delta x = 0000*, \Delta y = 0010*, \Delta g = 0000* \rightarrow \Delta h = 1010*)$. Let c_1 and c_2 be the carry bit vectors of $x \boxplus y$ and $z \boxplus g$ respectively. Then the Equations (12-14) and (16) can be still obtained. By Property 1, $\Delta z[3] = 1$ is obvious. Due to Equations (12) and (13), clearly $\Delta c_1[2] = 1$ and $\Delta c_2[2] = 1$.

According to the value of $\Delta z[4]$, there are two cases.

Case 1: Suppose $\Delta z[4] = 0$. Since $\Delta x[4] = 0, \Delta y[4] = 0, \Delta g[4] = 0, \Delta h[4] = 1$, we have $\Delta c_1[3] = 0$ and $\Delta c_2[3] = 1$ by Equation (14) and Equation (16) respectively. Since $\Delta x[3] = 0, \Delta y[3] = 0$, according to Equation (14), we have

$$\Delta c_1[3] = \Delta c_1[2] \wedge (x[3] \oplus y[3]) = 0 \Rightarrow x[3] \oplus y[3] = 0.$$

By Definition 1,

$$z[3] = x[3] \oplus y[3] \oplus c_1[2] = c_1[2].$$

Meanwhile, on the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, since $\Delta z[3] = 1, \Delta c_2[2] = 1, \Delta g[3] = 0$, by Equation (16), we have

$$\Delta c_2[3] = \Delta(z[3] \wedge c_2[2]) = 1 \Rightarrow z[3] = c_2[2].$$

Then, considering the value of $\Delta z[2]$, there are two sub-situations.

Subcase 1.1: When $\Delta z[2] = 1$, we have $\Delta c_1[1] = 0$ and $\Delta c_2[1] = 0$. Due to $\Delta x[2] = 0, \Delta y[2] = 1$ and $\Delta c_1[2] = 1$,

$$\Delta c_1[2] = \Delta y[2] \wedge (x[2] \oplus c_1[1]) = 1 \Rightarrow x[2] \oplus c_1[1] = 1.$$

By Definition 1,

$$\begin{aligned} c_1[2] &= (x[2] \wedge c_1[1]) \oplus (x[2] \wedge y[2]) \oplus (y[2] \wedge c_1[1]) \\ &= 0 \oplus (x[2] \oplus c_1[1]) \wedge y[2] = y[2], \end{aligned}$$

$$z[2] = x[2] \oplus y[2] \oplus c_1[1] = 1 \oplus y[2] = 1 \oplus c_1[2].$$

Thus,

$$z[3] = 1 \oplus z[2]. \tag{32}$$

Due to $\Delta z[2] = 1, \Delta g[2] = 0, \Delta c_2[1] = 0$, by Equation (16), we have

$$\Delta c_2[2] = \Delta z[2] \wedge (g[2] \oplus c_2[1]) = 1 \Rightarrow g[2] \oplus c_2[1] = 1.$$

By Definition 1,

$$\begin{aligned} c_2[2] &= (z[2] \wedge g[2]) \oplus (z[2] \wedge c_2[1]) \oplus (g[2] \wedge c_2[1]) \\ &= z[2] \wedge (g[2] \oplus c_2[1]) \oplus 0 = z[2]. \end{aligned}$$

Thus $z[3] = z[2]$ which is in contradiction to Equation (32).

Subcase 1.2: When $\Delta z[2] = 0$, we have $\Delta z[1] = 1$ by Property 1 (or Property 2) as $\Delta x[1] = 0$ and $\Delta y[1] = 0$ (or $\Delta g[1] = 0$ and $\Delta h[1] = 0$). Then, we obtain $\Delta c_1[1] = 1, \Delta c_1[0] = 1$ and $\Delta c_2[1] = 1, \Delta c_2[0] = 1$ by Equations (12) and (13). Similar to **Subcase 1**,

$$\begin{cases} \Delta c_1[2] = \Delta(y[2] \wedge c_1[1]) \Rightarrow c_1[2] = y[2] = c_1[1], \\ \Delta c_1[1] = \Delta c_1[0] \wedge (x[1] \oplus y[1]) \Rightarrow x[1] \oplus y[1] = 1 \text{ and } c_1[1] = c_1[0], \end{cases}$$
$$\Rightarrow z[1] = x[1] \oplus y[1] \oplus c_1[0] = 1 \oplus c_1[0] = 1 \oplus c_1[2].$$

Thus,

$$z[3] = 1 \oplus z[1]. \tag{33}$$

And,

$$\begin{cases} \Delta c_2[1] = \Delta(z[1] \wedge c_2[0]) \Rightarrow c_2[1] = z[1] = c_2[0], \\ \Delta c_2[2] = \Delta c_2[1] \wedge (z[2] \oplus g[2]) \Rightarrow z[2] \oplus g[2] = 1 \text{ and } c_2[2] = c_2[1], \end{cases}$$
$$\Rightarrow z[1] = c_2[2].$$

Thus $z[3] = z[1]$ which is in contradiction to Equation (33).

Case 2: Suppose $\Delta z[4] = 1$. Since $\Delta x[4] = 0, \Delta y[4] = 0, \Delta g[4] = 0, \Delta h[4] = 1$, we have $\Delta c_1[3] = 1$ and $\Delta c_2[3] = 0$ by Equation (14) and Equation (16) respectively. Similar to **Case 1**, there are two sub-situations according to the value of $\Delta z[2]$.

Subcase 2.1: When $\Delta z[2] = 1$, in the first modular addition $z = x \boxplus y$ and $z' = x' \boxplus y'$, we have

$$\begin{cases} z[3] = 1 \oplus c_1[2] \\ z[2] = 1 \oplus c_1[2] \end{cases} \Rightarrow z[3] = z[2]; \tag{34}$$

in the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, we have

$$\begin{cases} z[3] = 1 \oplus c_2[2] \\ z[2] = c_2[2] \end{cases} \Rightarrow z[3] = 1 \oplus z[2]. \tag{35}$$

Then Equation (34) and Equation (35) constitute a contradiction.

Subcase 2.2: When $\Delta z[2] = 0$, we have

$$\begin{cases} z[3] = 1 \oplus c_1[2] \\ z[1] = 1 \oplus c_1[2] \end{cases} \Rightarrow z[3] = z[1]; \tag{36}$$

and

$$\begin{cases} z[3] = 1 \oplus c_2[2] \\ z[1] = c_2[2] \end{cases} \Rightarrow z[3] = 1 \oplus z[1]. \tag{37}$$

Then Equation (36) and Equation (37) constitute a contradiction.

Thus, ($\Delta x = 0000*$, $\Delta y = 0010*, \Delta g = 0000* \nrightarrow \Delta h = 1010*$). □

B The proof of Property 8

Property 8. Let $z = x \boxplus y$, $z' = x' \boxplus y'$, $h = z \boxplus g$ and $h' = z' \boxplus g'$, where $x, y, z, g, h, x', y', z', g', h' \in \mathbb{F}_2^4$. Suppose that $\Delta x = x \oplus x'$, $\Delta y = y \oplus y'$, $\Delta z = z \oplus z'$, $\Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. Then

$$(\Delta x = 0000,\ \Delta y = 00**, \Delta g = 0000 \nrightarrow \Delta h = 100*) \tag{38}$$

Proof. `Reductio ad absurdum.` Suppose ($\Delta x = 0000, \Delta y = 00**, \Delta g = 0000 \to \Delta h = 100*$). Let c_1 and c_2 be the carry bit vector of $x \boxplus y$ and $z \boxplus g$ respectively. Then the Equations (12-14) and (16) can be still obtained. By Property 1, we have $\Delta z[2:1] = 11$. Due to Equation (12) and (13), clearly $\Delta c_1[1] = 1, \Delta y[1] \oplus \Delta c_1[0] = 1$ and $\Delta c_2[1] = 1, \Delta c_2[0] = 1$.

According to the value of $\Delta z[3]$, there are two cases.

Case 1: Suppose $\Delta z[3] = 1$. Since $\Delta x[3] = 0, \Delta y[3] = 0, \Delta g[3] = 0, \Delta h[3] = 0$, then $\Delta c_1[2] = 1$ and $\Delta c_2[2] = 0$ due to Equation (12) and Equation (13) respectively. By Equation (14) and $\Delta x[2] = 0, \Delta y[2] = 0, \Delta c_1[1] = 1$, we have

$$\Delta c_1[2] = \Delta c_1[1] \wedge (x[2] \oplus y[2]) = 1 \Rightarrow x[2] \oplus y[2] = 1.$$

By Definition 1,

$$z[2] = x[2] \oplus y[2] \oplus c_1[1] = 1 \oplus c_1[1].$$

Due to $\Delta x[1] = 0, \Delta c_1[1] = 1, \Delta y[1] \oplus \Delta \bar{c}[0] = 1$ and Equation (14), we have

$$\Delta c_1[1] = \begin{cases} \Delta(x[1] \wedge c_1[0]) \oplus \Delta(y[1] \wedge c_1[0]) \Rightarrow x[1] \oplus y[1] = 1, & \textit{if } \Delta y[1] = 0; \\ \Delta(x[1] \wedge y[1]) \oplus \Delta(y[1] \wedge c_1[0]) \Rightarrow x[1] \oplus c_1[0] = 1, & \textit{if } \Delta y[1] = 1. \end{cases}$$

Additionally, by Definition 1,

$$c_1[1] = \begin{cases} c_1[0], & \textit{if } \Delta y[1] = 0; \\ y[1], & \textit{if } \Delta y[1] = 1. \end{cases} \Rightarrow z[1] = \begin{cases} 1 \oplus c_1[0] = 1 \oplus c_1[1], & \textit{if } \Delta y[1] = 0; \\ 1 \oplus y[1] = 1 \oplus c_1[1], & \textit{if } \Delta y[1] = 1. \end{cases}$$

$$\Rightarrow z[1] = 1 \oplus c_1[1].$$

Since $z[2] = 1 \oplus c_1[1]$,

$$z[2] = z[1]. \tag{39}$$

Meanwhile, on the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, by Equation (16) and $\Delta z[2] = 1, \Delta c_2[1] = 1, \Delta g[2] = 0, \Delta z[1] = 1, \Delta c_2[0] = 1, \Delta g[1] = 0$, we have

$$\Delta c_2[2] = \Delta(z[2] \wedge c_2[1]) = 0 \Rightarrow z[2] = 1 \oplus c_2[1];$$

$$\Delta c_2[1] = \Delta(z[1] \wedge c_2[0]) = 1 \Rightarrow z[1] = c_2[0] = c_2[1].$$

Thus $z[2] = 1 \oplus z[1]$ which is in contradiction to Equation (39).

Case 2: Suppose $\Delta z[3] = 0$. Since $\Delta x[3] = 0, \Delta y[3] = 0, \Delta g[3] = 0, \Delta h[3] = 0$, then $\Delta c_1[2] = 0$ and $\Delta c_2[2] = 1$ due to Equation (12) and Equation (13) respectively. Similar to **Case 1**, in the first modular addition $z = x \boxplus y$ and $z' = x' \boxplus y'$, we have

$$\begin{cases} z[2] = c_1[1] \\ z[1] = 1 \oplus c_1[1] \end{cases} \Rightarrow z[2] = 1 \oplus z[1]; \tag{40}$$

in the second modular addition $h = z \boxplus g$ and $h' = z' \boxplus g'$, we have

$$\begin{cases} z[2] = c_2[1] \\ z[1] = c_2[1] \end{cases} \Rightarrow z[2] = z[1]. \tag{41}$$

Then Equation (40) and Equation (41) constitute a contradiction.

To sum up, $(\Delta x = 0000,\ \Delta y = 00**, \Delta g = 0000 \not\twoheadrightarrow \Delta h = 100*)$ □

C The proof of Property 9

Property 9. $x \boxplus y = z (\bmod 2^4), x' \boxplus y' = z' (\bmod 2^4), z \boxplus d = e (\bmod 2^4), z' \boxplus d' = e' (\bmod 2^4), f \boxplus g = h (\bmod 2^5)$ and $f' \boxplus g' = h' (\bmod 2^5)$. Suppose that $\Delta x = x \oplus x', \Delta y = y \oplus y', \Delta z = z \oplus z', \Delta d = d \oplus d', \Delta e = e \oplus e', \Delta f = f \oplus f', \Delta g = g \oplus g'$ and $\Delta h = h \oplus h'$. If $f[4:1] = e$, then

$$(\Delta x = 000*, \Delta y = 000*, \Delta d = 10**, \Delta g = 0000* \not\twoheadrightarrow \Delta h = 0000*) \tag{42}$$

Proof. Reductio ad absurdum. Suppose $(\Delta x = 000*, \Delta y = 000*, \Delta d = 10**, \Delta g = 0000* \rightarrow \Delta h = 0000*)$. Firstly, by Property 1, we can obtain A is the set of the possible value of $\Delta z \| \Delta f[4:1]$ based on SAT/SMT solver, where

$$\begin{aligned} A = \{&01100000, 01100001, 01110000, 01110001, 11100000, 11100001, 11110000, \\ &11110001, 00000111, 00001111, 00010111, 00011111, 00100111, 00101111, \\ &00110111, 00111111, 01100011, 01100111, 01101111, 01110011, 01110111, \\ &01111111, 11100011, 11100111, 11101111, 11110011, 11110111, 11111111\}. \end{aligned}$$

Let $c_1, c_1', c_2, c_2' \in \mathbb{F}_2^4$ be the carry bit vectors used to calculate $z, z', e,$ and e', and $c_3, c_3' \in \mathbb{F}_2^5$ be the carry bit vectors used to calculate f, and f'. And $\Delta c_1 = c_1 \oplus c_1', \Delta c_2 = c_2 \oplus c_2' \Delta c_3 = c_3 \oplus c_3'$. By Definition 1, we have

$$\Delta z[i] = \Delta x[i] \oplus \Delta y[i] \oplus \Delta c_1[i-1] \tag{43}$$

$$\Delta e[i] = \Delta z[i] \oplus \Delta d[i] \oplus \Delta c_2[i-1] \tag{44}$$

$$\Delta h[i] = \Delta f[i] \oplus \Delta g[i] \oplus \Delta c_3[i-1] \tag{45}$$

$$\Delta c_1[i] = \Delta(x[i] \wedge y[i]) \oplus \Delta(x[i] \wedge c_1[i-1]) \oplus \Delta(y[i] \wedge c_1[i-1]) \tag{46}$$

$$\Delta c_2[i] = \Delta(z[i] \wedge d[i]) \oplus \Delta(z[i] \wedge c_2[i-1]) \oplus \Delta(d[i] \wedge c_2[i-1]) \tag{47}$$

$$\Delta c_3[i] = \Delta(f[i] \wedge g[i]) \oplus \Delta(f[i] \wedge c_3[i-1]) \oplus \Delta(g[i] \wedge c_3[i-1]) \tag{48}$$

According to the value of $\Delta z \| \Delta f[4:1]$, there are three cases.

(1) When $\Delta e[3:1] = \Delta f[4:2] = 000$, by Property 1, $\Delta z[2:1] = 11$ is obvious. Due to Equation (43) and (44), $\Delta c_1[1] = 1, \Delta c_1[0] = 1$ and $\Delta c_2[1] = 1, d[1] = 1 \oplus \Delta c_2[0]$. By Definition 1 and (46),

$$\Delta c_1[1] = \Delta c_1[0] \wedge (x[1] \oplus y[1]) = 1 \Rightarrow x[1] \oplus y[1] = 1$$
$$\Rightarrow z[1] = 1 \oplus c_1[0] = 1 \oplus c_1[1].$$

$$\begin{cases} \Delta c_1[2]{=}\Delta c_1[1]\wedge(x[2] \oplus y[2]){=}1 \Rightarrow x[2] \oplus y[2]{=}1, & \textit{if } \Delta z[3]{=}1; \\ \Delta c_1[2]{=}\Delta c_1[1]\wedge(x[2] \oplus y[2]){=}0 \Rightarrow x[2] \oplus y[2]{=}0, & \textit{if } \Delta z[3]{=}0. \end{cases}$$

$$z[2] = x[2] \oplus y[2] \oplus c_1[1] = \begin{cases} 1 \oplus c_1[1]{=}z[1], & \textit{if } \Delta z[3]{=}1; \\ c_1[1]{=}1 \oplus z[1], & \textit{if } \Delta z[3]{=}0. \end{cases}$$

However,

$$\begin{cases} \Delta c_2[1]{=}\Delta(z[1]\wedge c_2[0]){=}1 \Rightarrow z[1]{=}c_2[0] = c_2[1], & \textit{if } \Delta c_2[0]{=}1; \\ \Delta c_2[1]{=}\Delta(z[1]\wedge d[1]){=}1 \Rightarrow z[1]{=}d[1] = c_2[1], & \textit{if } \Delta c_2[0]{=}0. \end{cases} \Rightarrow z[1] = c_2[1].$$

and

$$\begin{cases} \Delta c_2[2]{=}\Delta(z[2]\wedge c_2[1]){=}0 \Rightarrow z[2]{=}1 \oplus c_2[1]{=}1 \oplus z[1], & \textit{if } \Delta z[3]{=}1; \\ \Delta c_2[2]{=}\Delta(z[2]\wedge c_2[1]){=}1 \Rightarrow z[2]{=}c_2[1]{=}z[1], & \textit{if } \Delta z[3]{=}0. \end{cases}$$

(2) When $\Delta z[3:2] = 00, \Delta e[2:0] = \Delta f[3:1] = 111$, clearly $\Delta c_2[1] = 1, \Delta c_3[2] = \Delta c_3[1] = \Delta c_3[0] = 1$ due to Equations (47) and (48). By Equation (48),

$$\Delta c_3[2] = \Delta(f[2] \wedge c_3[1]) \Rightarrow f[2] = c_3[1] = c_3[2].$$

$$\Delta c_3[1] = \Delta(f[1] \wedge c_3[0]) \Rightarrow f[1] = c_3[0] = c_3[1].$$

Thus,

$$f[2] = f[1] = c_3[2] \tag{49}$$

Subcase 1: If $\Delta e[3] = \Delta f[4] = 0$, then $\Delta c_3[3] = 0$ and $\Delta c_2[2] = 1$. By Equation (48),

$$\Delta c_3[3] = \Delta(f[3] \wedge c_3[2]) = 0 \Rightarrow f[3] = 1 \oplus c_3[2].$$

Thus

$$f[3] = 1 \oplus f[2] = 1 \oplus f[1]. \tag{50}$$

By Equation (47) and Definition 1, since $\Delta z[2] = 0, \Delta d[2] = 0, \Delta c_2[2] = 1$ and $\Delta c_2[1] = 1$, we have

$$\Delta c_2[2] = \Delta c_2[1] \wedge (z[2] \oplus d[2]) \Rightarrow z[2] \oplus d[2] = 1 \Rightarrow e[2] = 1 \oplus c_2[1]. \tag{51}$$

By Equation (44),since$\Delta e[1] = 1$, we have $\Delta z[1] \oplus \Delta d[1] \oplus c_2[0] = 1$.

a) If $\Delta c_2[0] = 0$, then by Equation (47),

$$\begin{cases}\Delta c_2[1]{=}\Delta z[1]\wedge(d[1]\oplus c_2[0]){=}1, & \textit{if } \Delta z[1]{=}1;\\ \Delta c_2[1]{=}\Delta d[1]\wedge(z[1]\oplus c_2[0]){=}1, & \textit{if } \Delta z[1]{=}0.\end{cases}$$

$$\Rightarrow \begin{cases}d[1]\oplus c_2[0]{=}1, & \textit{if } \Delta z[1]{=}1;\\ z[1]\oplus c_2[0]{=}1, & \textit{if } \Delta z[1]{=}0.\end{cases} \Rightarrow \begin{cases}c_2[1]{=}z[1], & \textit{if } \Delta z[1]{=}1;\\ c_2[1]{=}d[1], & \textit{if } \Delta z[1]{=}0.\end{cases}$$

Thus, by Definition 1, we have

$$\begin{cases}e[1]{=}z[1]\oplus d[1]\oplus c_2[0] = 1\oplus z[1] = 1\oplus c_2[1], & \textit{if } \Delta z[1]{=}1;\\ e[1]{=}z[1]\oplus d[1]\oplus c_2[0] = 1\oplus d[1] = 1\oplus c_2[1], & \textit{if } \Delta z[1]{=}0.\end{cases} \tag{52}$$
$$\Rightarrow e[1] = 1\oplus c_2[1].$$

Combining Equation (51) and (52), $e[1] = e[2]$, i.e.

$$f[2] = f[3]. \tag{53}$$

Equation (50) and Equation (53) constitute a contradiction.

b) If $\Delta c_2[0] = 1$, then by Equation (47), when $\Delta z[1] = \Delta d[1] = 0$

$$\Delta c_2[1] = \Delta c_2[0]\wedge(z[1]\oplus d[1]) = 1 \Rightarrow z[1]\oplus d[1] = 1 \Rightarrow c_2[1] = c_2[[0].$$

Thus, $e[1] = z[1]\oplus d[1]\oplus c_2[0] = 1\oplus c_2[0] = 1\oplus c_2[1]$. According to Equation (51), $e[1] = e[2]$, then we obtain Equation (53) again which is in contradiction to Equation (50). When $\Delta z[1] = \Delta d[1] = 1 = \Delta c_2[0]$, we can obtain that $d[1]\oplus c_2[1] = d'[1]\oplus c_2'[1]$.

$$\begin{cases}\Delta c_2[1]{=}\Delta z[1]\wedge(d[1]\oplus c_2[0])\Rightarrow c_2[1] = z[1], & \textit{if } d[1]{\oplus} c_2[0] = 1;\\ \Delta c_2[1]{=}\Delta(d[1]{\wedge}c_2[0]){\Rightarrow}c_2[0]{=}d[1] = d_2[1]\text{and}\Delta d[1]{=}1{=}\Delta c_2[0], & \textit{if } d[1]{\oplus} c_2[0] = 0.\end{cases}$$

By Definition 1, $e[1] = 1\oplus z[1] = 1\oplus c_2[1]$ when $d[1]\oplus c_2[0] = 1$. According to Equation (51), $e[1] = e[2]$, then we obtain Equation (53) again which is in contradiction to Equation (50). For the case $d[1]\oplus c_2[0] = 0$, since $e[0] = z[0]\oplus d[0], c_2[0] = z[0]\wedge d[0], \Delta e[0] = 1, \Delta c[0] = 1$, we obtain

$$\begin{cases}\Delta e[0]{=}\Delta d[0], \Delta c_2[0]{=}\Delta d[0] \Rightarrow e[0]{=}1\oplus d[0]{=}1\oplus c_2[0]{=}1\oplus c_2[1], & \textit{if } \Delta d[0] = 1;\\ \Delta e[0]{=}\Delta z[0], \Delta c_2[0]{=}\Delta z[0] \Rightarrow e[0]{=}1\oplus z[0]{=}1\oplus c_2[0]{=}1\oplus c_2[1], & \textit{if } \Delta d[0] = 0.\end{cases}$$
$$\Rightarrow e[0] = 1\oplus c_2[1]. \tag{54}$$

According to Equation (51), we have

$$e[2] = e[0], \textit{ i.e. } f[3] = f[1]. \tag{55}$$

Equation (50) and Equation (55) constitute a contradiction.

Subcase 2: The reason for the contradiction when $\Delta e[3] = \Delta f[4] = 1$ is similar to the **Subcase 1**.

(3) For other cases in set A, using similar discussions, there would be contradictions between $z[2]$ and $z[1]$ or among $f[3], f[2]$ and $f[1]$ as well.

□

D Differentials with probability 1 in LEA

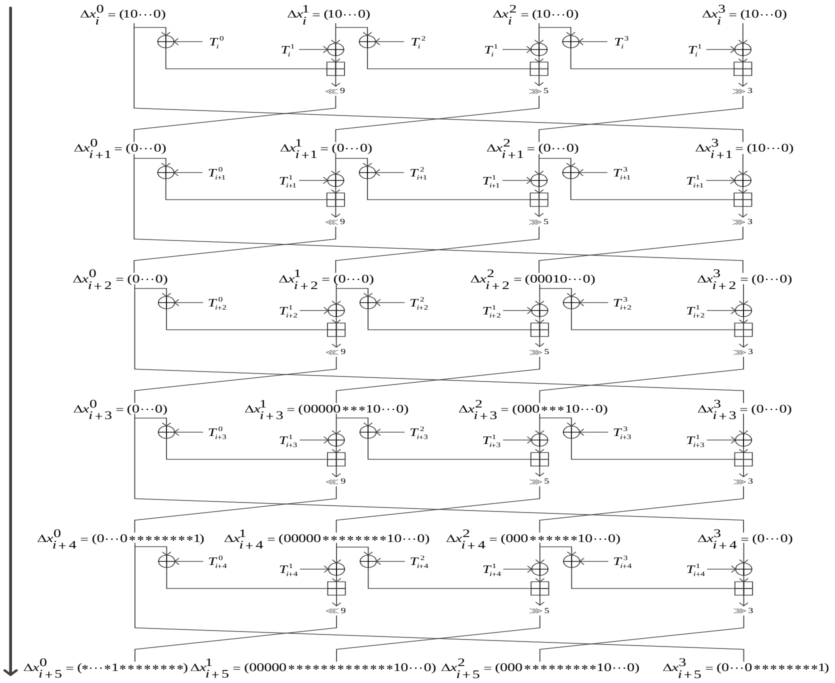


Figure 9: first 5 rounds of 11-round ID for LEA-128

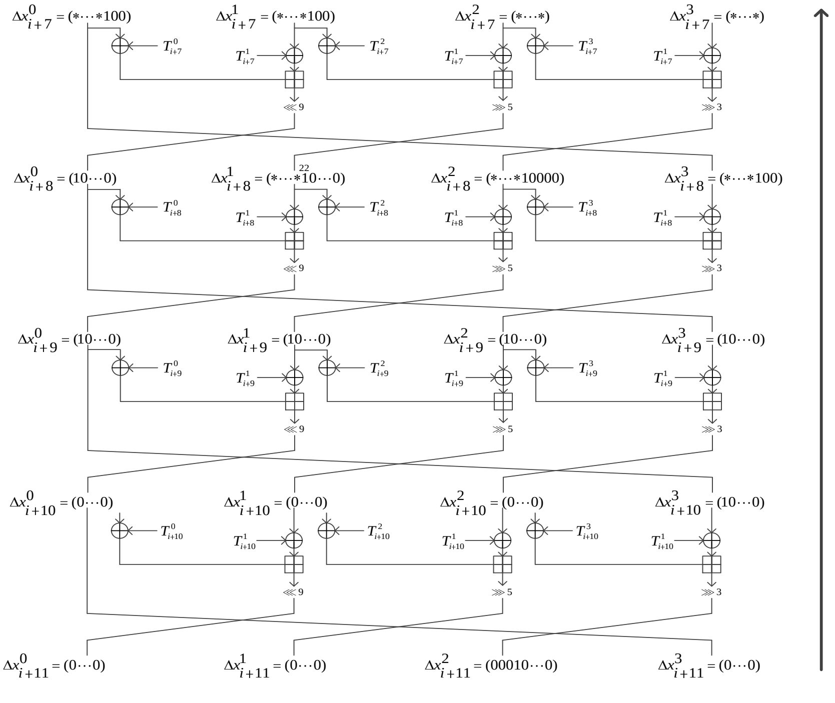


Figure 10: last 4 rounds of 11-round ID for LEA-128

E Differentials with probability 1 in CHAM-64/28

Table 9: first 10 rounds of 22-round ID for CHAM-64/128

	r	Δx_r	Δx_r	r
x^0	i	0000000010000000	10000000********	$i+9$
x^1		1000000000000000	0000000000000000	
x^2		0000000000000000	******10********	
x^3		0000000000000000	**************1*	
x^0	$i+1$	1000000000000000	0000000000000001	$i+8$
x^1		0000000000000000	10000000********	
x^2		0000000000000000	0000000000000000	
x^3		0000000000000000	******10********	
x^0	$i+2$	0000000000000000	0000000000000000	$i+7$
x^1		0000000000000000	0000000000000001	
x^2		0000000000000000	10000000********	
x^3		0000000010000000	0000000000000000	
x^0	$i+3$	0000000000000000	0000000000000000	$i+6$
x^1		0000000000000000	0000000000000000	
x^2		0000000010000000	0000000000000010	
x^3		0000000000000000	10000000********	
x^0	$i+4$	0000000000000000	0000000010000000	$i+5$
x^1		0000000010000000	0000000000000000	
x^2		0000000000000000	0000000000000000	
x^3		0000000000000000	0000000000000001	

Table 10: last 5 rounds of 22-round ID for CHAM-64/128

	r	Δx_r
x^0	$i+18$	**10000000000000
x^1		0000000000000000
x^2		0000000000000000
x^3		0000000000000000
x^0	$i+19$	0000000000000000
x^1		0000000000000000
x^2		0000000000000000
x^3		0100000000000000
x^0	$i+20$	0000000000000000
x^1		0000000000000000
x^2		0100000000000000
x^3		0000000000000000
x^0	$i+21$	0000000000000000
x^1		0100000000000000
x^2		0000000000000000
x^3		0000000000000000
x^0	$i+22$	0100000000000000
x^1		0000000000000000
x^2		0000000000000000
x^3		0000000010000000

F Mid 9 rounds for 22-round ID for CHAM-64/128

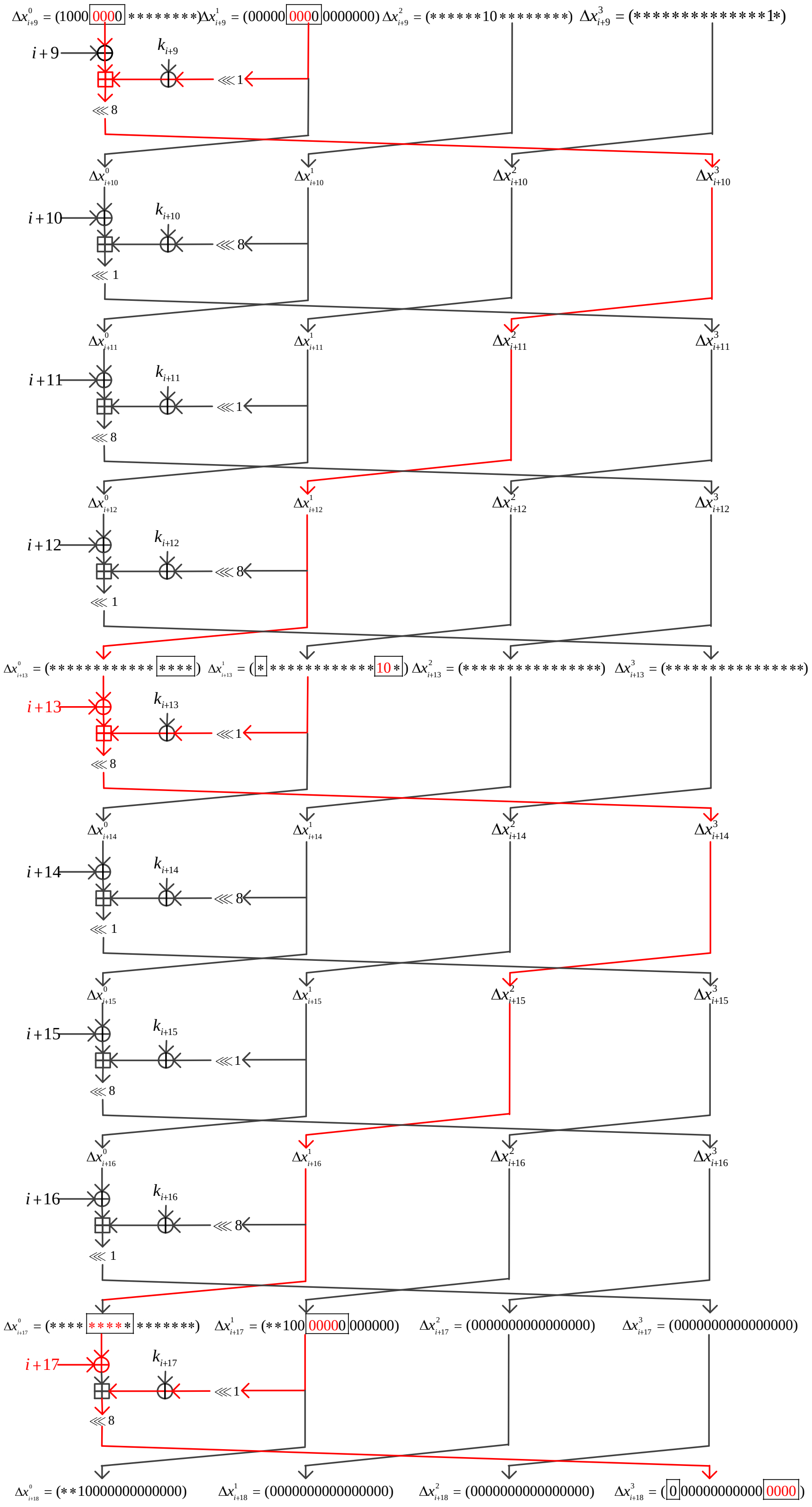

Δx^0_{i+9} = (1000 $\boxed{0000}$ ********)
Δx^1_{i+9} = (00000 $\boxed{0000}$ 0000000)
Δx^2_{i+9} = (******10********)
Δx^3_{i+9} = (**************1*)
$i+9$
k_{i+9}
⋘ 1
⋘ 8
Δx^0_{i+10}
Δx^1_{i+10}
Δx^2_{i+10}
Δx^3_{i+10}
$i+10$
k_{i+10}
⋘ 8
⋘ 1
Δx^0_{i+11}
Δx^1_{i+11}
Δx^2_{i+11}
Δx^3_{i+11}
$i+11$
k_{i+11}
⋘ 1
⋘ 8
Δx^0_{i+12}
Δx^1_{i+12}
Δx^2_{i+12}
Δx^3_{i+12}
$i+12$
k_{i+12}
⋘ 8
⋘ 1
Δx^0_{i+13} = (************ $\boxed{****}$)
Δx^1_{i+13} = ($\boxed{*}$ ************ $\boxed{10*}$)
Δx^2_{i+13} = (****************)
Δx^3_{i+13} = (****************)
$i+13$
k_{i+13}
⋘ 1
⋘ 8
Δx^0_{i+14}
Δx^1_{i+14}
Δx^2_{i+14}
Δx^3_{i+14}
$i+14$
k_{i+14}
⋘ 8
⋘ 1
Δx^0_{i+15}
Δx^1_{i+15}
Δx^2_{i+15}
Δx^3_{i+15}
$i+15$
k_{i+15}
⋘ 1
⋘ 8
Δx^0_{i+16}
Δx^1_{i+16}
Δx^2_{i+16}
Δx^3_{i+16}
$i+16$
k_{i+16}
⋘ 8
⋘ 1
Δx^0_{i+17} = (**** $\boxed{*****}$ *******)
Δx^1_{i+17} = (**100 $\boxed{00000}$ 000000)
Δx^2_{i+17} = (0000000000000000)
Δx^3_{i+17} = (0000000000000000)
$i+17$
k_{i+17}
⋘ 1
⋘ 8
Δx^0_{i+18} = (**10000000000000)
Δx^1_{i+18} = (0000000000000000)
Δx^2_{i+18} = (0000000000000000)
Δx^3_{i+18} = ($\boxed{0}$ 00000000000 $\boxed{0000}$)

Figure 11: Contradiction on the $i+9$-th $\sim$ $i+18$-th rounds of 22-round ID for CHAM-64/128