Revisiting Boomerang Attacks on Lightweight ARX and AND-RX Ciphers with Applications to KATAN, SIMON and CHAM

Li Yu[a], Je Sen Teh[b,c,*]

[a] *CRISES research group, University Rovira i Virgili, Tarragona, Spain*
[b] *Deakin Cyber Research and Innovation Centre, Deakin University, Geelong, Australia*
[c] *School of Information Technology, Deakin University, Geelong, Australia*

Abstract

In this paper, we investigate the security of lightweight block ciphers, focusing on those that utilize the ADD-Rotate-XOR (ARX) and AND-Rotate-XOR (AND-RX) design paradigms. More specifically, we examine their resilience against boomerang-style attacks. First, we propose an automated search strategy that leverages the boomerang connectivity table (BCT) for AND operations ($\wedge BCT$) to conduct a complete search for boomerang and rectangle distinguishers for AND-RX ciphers. The proposed search strategy automatically considers all possible $\wedge BCT$ switches in the middle of the boomerang to optimise distinguishing probability. The correctness of the search strategy was verified experimentally. We were able to find the best boomerang and rectangle distinguishers to date in the single-key model for lightweight block ciphers KATAN32/48/64 and SIMON32/48. Next, we investigated BCT properties of ARX ciphers and discovered that a truncated boomerang switch could be formulated for the lightweight ARX cipher, CHAM. We were able to find the best single-key and related-key rectangle distinguishers to date for CHAM. Our findings provide more accurate security margins of these lightweight ciphers against boomerang-style attacks.[1]

*Corresponding author
Email address: j.teh@deakin.edu.au (Je Sen Teh)

[1] This postprint corrects two minor typographical errors in the published manuscript (https://doi.org/10.1016/j.jisa.2024.103950) – the labelling of rotations in Figure 6 (pg. 28) and the ABCT switch pattern in the 41-round related-key rectangle distinguisher (pg. 31).

Keywords: ARX, Block ciphers, Boomerang attack, Boomerang switch, Cryptanalysis, Rectangle attack.

1. Introduction

The Internet of Things (IoT) has become integral to modern life, significantly enhancing efficiency and productivity across various sectors such as agriculture [1] and healthcare [2]. IoT systems comprise interconnected devices and sensors, most of which have limited computational capabilities, that transfer vast amounts of data over the network. Therefore, lightweight encryption algorithms are essential to secure communication between these devices [3]. The National Institute of Standards and Technology (NIST) recently completed its lightweight encryption standardization efforts and announced the decision to standardize the Ascon family [4].

Various cryptanalysis techniques were employed to evaluate the security of the lightweight encryption candidates throughout the NIST standardization efforts, including differential cryptanalysis. An attacker aims to find differential trails (the propagation of a plaintext difference to a ciphertext difference) that occur with high probability. These trails can then be used as statistical distinguishers in key recovery attacks. The boomerang attack [5] is an extension of the classical differential attack. Rather than finding a long differential trail, two shorter trails can be concatenated to form a longer distinguisher. Further refinements to the boomerang framework were later proposed that included the amplified boomerang and rectangle attacks [6]. In rectangle attacks, one can consider multiple valid transitions between the concatenated trails to further improve the distinguishing probability. However, the transition from one trail to another in both the boomerang and rectangle attacks is probabilistic. The aforementioned transition can be formally analysed using the sandwich framework [7] that divides a cipher into three parts, $E = E_1 \circ E_m \circ E_0$, where the dependency between E_0 and E_1 is addressed by analyzing the probability of E_m.

Cid et al. [8] later introduced a new tool called the *Boomerang Connectivity Table* (BCT) that allows to systematically evaluate E_m. They showed how BCT can be used when E_m covers one round of a substitution-permutation network (SPN) and also briefly described how BCT can be calculated for modular addition. The BCT framework was later refined to cover multiple rounds of SPN [9, 10] and Feistel constructions [11]. These findings

were used in some of the best boomerang and rectangle attacks on AES [9], SKINNY [12], CRAFT [12], WARP [13, 14] and CLEFIA [15]. Most of the research work on BCT has focused on block ciphers with S-boxes as their primary source of nonlinearity. More recently, there has been research looking into the use of the BCT framework to perform boomerang attacks on ADD-Rotate-XOR (ARX) ciphers such as SPECK and LEA [16] as well as AND-RX ciphers like SIMON and KATAN [17].

Our Contributions. Our work further investigates the application of BCT on both ARX and AND-RX ciphers. We first utilise the AND ($\wedge$) BCT or simply $\wedge BCT$ in an SMT-based automated approach for deriving complete boomerang and rectangle distinguishers without having to individually compute E_0, E_m, and E_1 trails. For simplicity's sake, we refer to the search as an automated boomerang search, which also encompasses the rectangle search. The automated boomerang search was applied to KATAN and SIMON in the single-key setting. For the KATAN family of ciphers, we found the best[2] single-key rectangle distinguishers with up to 86, 83 and 62 rounds for the 32, 48 and 64-bit variants respectively. The correctness of the proposed automated boomerang search was experimentally verified on KATAN32. For the SIMON family of ciphers, we found the best single-key rectangle distinguishers with up to 13 and 16 rounds for the 32 and 48-bit variants respectively. Next, we investigated the BCT for modular addition (which we will refer to as ABCT) and applied it to find the best boomerang and rectangle distinguishers for CHAM64. Rather than an automated search, we rely on the properties of a truncated boomerang switch to find the best single-key rectangle distinguisher with up to 39 rounds and the best related-key boomerang distinguisher with up to 46 rounds. Our findings are summarised in Tables 1, 2 and 3.

Code Repository. All codes related to this paper are publicly available at github.com/boomerangas/ARX_PAPER.

Paper Outline. Section 2 introduces the boomerang attack and its switching effect before describing the target ciphers. Next, we describe the boomerang connectivity tables for AND and modular addition operations in Sec-

[2]The *best* distinguisher in our context is determined based on the number of rounds.

Table 1: Summary of Boomerang-style Distinguishers for KATAN in the Single-key Setting

Cipher	Word Size	Rounds	α	δ	w	w'	r_0	r_m	r_1	
KATAN	32	83	(0000,8081)	(0080,1081)	100	74.24	35		48	[18]
		83	(0000,8801)	(0010,0210)	48	28.65	39	4	40	Ours
		84	(8000,4400)	(0020,0420)	44	27.35	40	4	40	
		85	(1004,2080)	(0080,1080)	44	27.91	41	4	40	
		86	(1004,2080)	(0080,1081)	44	30.06	41	4	41	
	48	60	(0000,0090,4000)	0004,0200,0000)	38	21.52	35		25	[18]
		60	(0000,0402,0000)	(0002,0100,0000)	38	17.51	29	3	28	Ours
		81	(0000,0090,4000)	(2000,0000,0048)	60	39.39	39	3	39	
		82	(0000,0090,4000)	(1008,0000,0004)	64	38.37	40	3	39	
		83	(0000,0090,4000)	(0090,4000,000D)	66	41.27	40	3	40	
	64	56	(0000,0000,0400,2001)	(0020,1100,8000,0000)	88	67.54	30		26	[18]
		56	(0000,0010,0080,0400)	(0020,3100,8000,0000)	56	51.92	27	3	26	Ours
		60	(2012,0844,0200,0000)	(0203,1008,0000,0044)	64	54.39	29	3	28	
		61	(8048,2010,0800,0000)	(0080,4402,0000,0051)	66	55.87	29	3	29	
		62	(0000,0020,0500,2001)	(2011,0080,0000,0040)	68	62.25	30	3	29	

Note: w and w' denote boomerang and rectangle weights respectively while r denotes the number of rounds for each part of the distinguisher. Probability is calculated as 2^{-w}.

Table 2: Summary of Boomerang-style Distinguishers for SIMON in the Single-key Setting

Cipher	Word Size	Rounds	α	δ	w	w'	r_0	r_m	r_1	
SIMON	32	13	(0000,0040)	(4000,0000)	36 (non-boomerang)	30.22	13			[19]
			(0010,0044)	(1100,0400)	44	28.54	6	1	6	Ours
	48	15	(0101,0004,4040)	(4440,4010,0000)	50 (non-boomerang)	43.01	15			[19]
			(0000,4000,4111)	(0006,4000,0100)	60	41.87	7	1	7	Ours
		16	(0400,0019,1000)	(2000,8280,0000)	86	46.45	7	1	8	

Note: w and w' denote boomerang and rectangle weights respectively while r denotes the number of rounds for each part of the distinguisher. Probability is calculated as 2^{-w}.

tion 3 and Section 4 respectively. The proposed automated boomerang search is discussed in Section 5. Results for AND-RX ciphers KATAN and SIMON, and ARX cipher CHAM are detailed in Section 6, Section 7 and Section 8 respectively. We conclude the paper in Section 9.

2. Preliminaries

2.1. Boomerang-style Attacks

The boomerang attack is a variant of differential cryptanalysis. The general motivation behind the boomerang attack is to construct longer distinguishers more effectively by combining two shorter differential trails, each of which can be searched independently. Wagner first introduced this attack and applied it to COCONUT98 [5].

Table 3: Summary of Boomerang-style Distinguishers for CHAM64

Setting	Rounds	Split	Input Diff.	Output Diff.	r	w'
SK	36	E_0	(8200,0100,0001,8000)	(0400,0004,****,****)	17	47.13
		E_1	(8200,****,****,8000)	(0004,0502,0088,0000)	18	
	37	E_0	(8200,0100,0001,8000)	(0004,0502,****,****)	18	52.57
		E_1	(9000,****,****,4000)	(0100,0281,0002,0000)	18	
	38	E_0	(8004,4082,8200,0100)	(0400,0004,****,****)	19	53.76
		E_1	(8200,****,****,8000)	(0004,0502,0088,0000)	18	
	39	E_0	(8004,4082,8200,0100)	(0400,0004,****,****)	19	59.27
		E_1	(8200,****,****,8000)	(0502,0088,0000,000A)	19	
RK	41	E_0	(8080,4040,4040,000)	(4200,0000,****,****)	20	49.28
		E_1	(8400,****,****,0000)	(0000,0400,0105,8080)	20	
	42	E_0	(8080,4040,4040,0000)	(4200,0000,****,****)	20	53.28
		E_1	(8400,****,****,0000)	(0400,0105,8080,0100)	21	
	43	E_0	(8080,4040,4040,0000)	(0000,0000,****,****)	21	52.24
		E_1	(****,****,****,****)	(0105,8080,0300,0B82)	21	
	44	E_0	(8080,4040,4040,0000)	(0000,0000,****,****)	21	59.88
		E_1	(****,****,****,****)	(8080,0300,0B82,030B)	22	
	46	E_0	(8080,4040,4040,0000)	(4200,0000,0000,0084)	20	62.78
		E_m	(4200,0000,0000,0084)	(0000,0000,0000,8401)	7	
		E_1	(0000,0000,0000,8401)	(0400,0105,8080,0100)	19	

[I] Except for the 46-round boomerang distinguisher, all experiments listed in this table involve the use of a 1-round boomerang switch ($r_m = 1$).

[II] For **Setting**, SK and RK denote the single-key setting and related-key setting respectively.

[III] The key difference for RK is Δ_{E_0}=(0000,0000,0000,0000,0000,4000,4000,0000) and $\Delta_{E_1} = (0080, 0000, 0000, 8000, 0000, 0000, 0000, 0000)$

[IV] Rectangle probability is calculated as $2^{-w'}$.

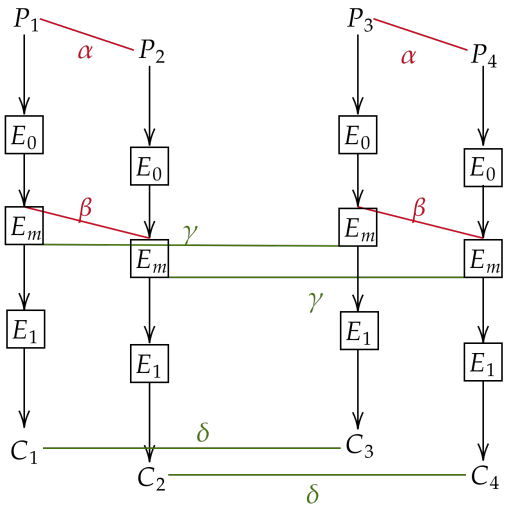

Figure 1: The structure of a boomerang (sandwich) attack.

The boomerang framework requires a quartet of chosen plaintexts, namely P_1, P_2, P_3, and P_4, along with their corresponding ciphertexts C_1, C_2, C_3, and C_4. The encryption operation is denoted as $E(\cdot)$. The primary objective of the boomerang attack is to decompose the cipher into two parts, $E = E_1 \circ E_0$, where E_0 represents the first half of the cipher and E_1 represents the last half. This process involves four specific differential trails: $\alpha \to \beta$ for E_0; $\gamma \to \delta$ for E_1; $\delta \to \gamma$ for E_1^{-1}; and $\beta \to \alpha$ for E_0^{-1}. Initially, a chosen plaintext pair P_1 and P_2 is encrypted in the forward direction $(E_1 \circ E_0)$. The resulting pair of ciphertexts C_1 and C_2 are then utilized to derive C_3 and C_4 based on the output difference, δ. These derived ciphertexts are decrypted $(E_0^{-1} \circ E_1^{-1})$ to obtain P_3 and P_4. A successful boomerang quartet is achieved when the condition $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$ is satisfied, indicating that the initial input difference "returns" intact after traversing the entire cipher, analogous to how a boomerang returns to its thrower.

The key objective of a boomerang attack is to identify a boomerang distinguisher that holds (or returns) with a high probability, enabling efficient key recovery similar to a classical differential attack. One main difference is that the boomerang attack employs quartets of chosen plaintexts for encryption rather than pairs. The probability p of a right quartet is determined by the probabilities p_0 and p_1 of the differential trails $\alpha \to \beta$ for E_0 and $\gamma \to \delta$ for E_1, respectively, as expressed by $p = p_0^2 p_1^2$.

To facilitate the analysis of how likely a boomerang distinguisher holds, an improvement to the boomerang framework called the sandwich attack was proposed [7]. This framework is depicted in Figure 1, where there is a transition round between E_0 and E_1 called the boomerang switch E_m. The distinguisher now has three distinct components, $E = E_1 \circ E_m \circ E_0$, where the effect of the boomerang switch is represented by $\beta \to \gamma$ and its associated probability is represented by r. Consequently, the overall probability of the boomerang distinguisher can be expressed as $p = p_0^2 p_1^2 r$. The following section discusses the boomerang switch in detail.

2.2. The Boomerang Switch and Boomerang Connectivity Tables

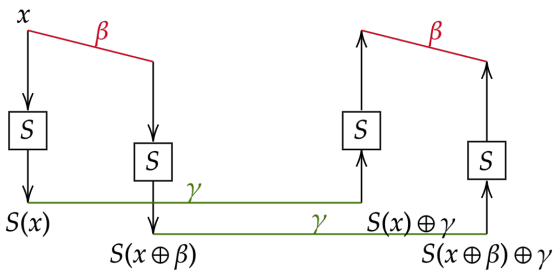

Figure 2: The 1-round boomerang switch for S-box ciphers.

In 2011, Murphy showed that independently chosen E_0 and E_1 trails may turn out to be incompatible and the boomerang never "returns" [20]. Cid et al. [8] provided a solution to the problem in the form of a new tool called the *Boomerang Connectivity Table* (BCT) that aids cryptanalysts in constructing valid boomerang distinguishers. They examined the boomerang switching behaviour of a partial round consisting only of one S-box layer that connects the upper and lower segments of the boomerang as illustrated in Figure 2. The formal definition of the BCT for substitution-permutation networks (SPN) is given by:

Definition 1. *((SPN) BCT [8]). Let $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an invertible function, and $\beta, \gamma \in \mathbb{F}_2^n$. The Boomerang Connectivity Table (BCT) of S can be described by a $2^n \times 2^n$ table, in which the entry for (β, γ) is computed by:*

$$BCT(\beta, \gamma) \quad = \quad \#\{x \in \mathbb{F}_2^n | S^{-1}(S(x) \oplus \gamma) \oplus S^{-1}(S(x \oplus \beta) \oplus \gamma) = \beta\}.$$

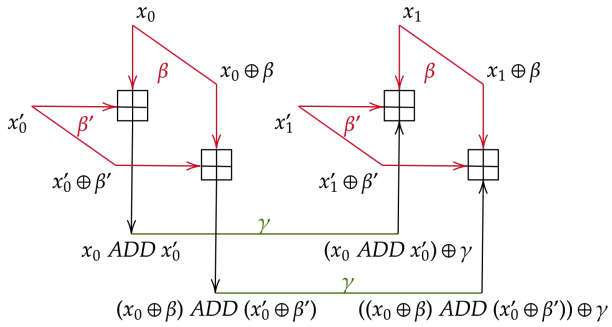

Figure 3: Illustration of the valid boomerang switch based on modular addition.

In the same paper, the authors also define the BCT for ADD ($(\beta, \beta', \gamma) \in \mathbb{F}_2^n | \beta \boxplus \beta' = \gamma$). Let $((x_1, x'_1), (x_2, x'_2), (x_3, x'_3), (x_4, x'_4))$ be a quartet of ADD inputs, where $x_1 \oplus x_2 = x_3 \oplus x_4 = \beta$, $x'_1 \oplus x'_2 = x'_3 \oplus x'_4 = \beta'$. This structure is illustrated in Figure 3. The output quartet is denoted by (y_1, y_2, y_3, y_3), where $y_1 \oplus y_3 = y_2 \oplus y_4 = \gamma$. The BCT for ADD is then defined as:

$$\begin{aligned} ABCT(\beta, \beta', \gamma) \quad = \quad & \#\{(x, x') \in \mathbb{F}_2^n | ((x \boxplus x') \oplus \gamma \boxminus x') \oplus \\ & (((x \oplus \beta) \boxplus (x' \oplus \beta') \oplus \gamma) \boxminus (s' \oplus \beta')) = \beta\}. \end{aligned}$$

They found that the ABCT has a similar *ladder switch* property as BCT but does not have the equivalent of an *S-box switch*. Instead, it has what the authors refer to as the *most significant bit (MSB) switch*. The authors note that one of the addends has to be fixed to make ADD operation invertible, i.e., $x'_1 = x'_3$ and $x'_2 = x'_4$.

Boukerrou et al. [11] later introduced the Feistel counterpart of the BCT known as the Feistel Boomerang Connectivity Table (FBCT). Since we do not need to consider bijective S-boxes for Feistel structures, FBCT can be used for any S-box from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ even when $n \neq m$. The formal definition of FBCT is given by:

Definition 2. *(FBCT [11]). Let $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $\alpha, \gamma \in \mathbb{F}_2^n$. The Feistel Boomerang Connectivity Table (FBCT) of S can be described by a $2^n \times 2^n$ table, in which the entry for (β, γ) is computed by:*

$$\begin{aligned} FBCT(\beta, \gamma) \quad = \quad & \#\{x \in \mathbb{F}_2^n | S(x) \oplus S(x \oplus \beta) \oplus \\ & (S(x \oplus \gamma) \oplus (S(x \oplus \beta \oplus \gamma) = 0\}. \end{aligned}$$

To evaluate the boomerang switch over more than one round, Wang and Peyrin [10] introduced the concept of the Boomerang Difference Table (BDT), a variant of the BCT that fixes additionally the S-box output difference of the upper trail. The same idea was concurrently investigated by Song et al. [9] as the upper BCT (UBCT) along with its counterpart, the lower BCT (LBCT) that can be used for the lower trail. We omit their definitions in this paper as we will not use these difference tables. In the remainder of the paper, we will refer to bitwise AND as $\wedge$, and addition and subtraction modulo 2^n as ADD and SUB respectively.

2.3. KATAN *Specification*

The KATAN family cipher is a hardware-oriented block cipher that was first proposed in CHES 2009 [21]. The KATAN cipher adopts an AND-RX structure that contributes to its high-performance characteristics in hardware environments. It has 254 rounds and three variants – KATAN32, KATAN48, and KATAN64, each designed to operate on 32-bit, 48-bit, and 64-bit blocks respectively. All variants have an 80-bit key that will be expanded by a key scheduling algorithm into 508 subkey bits. Suppose a key k is 80-bit and k_i represents the i-th bit in k. We then calculate subkey bits as:

$$sk_i = \begin{cases} k_i & for \quad i = 0 \dots 79 \\ k_{i-80} \oplus k_{i-61} \oplus k_{i-50} \oplus k_{i-13} & Otherwise \end{cases} \tag{1}$$

KATAN's round function divides the plaintext into two parts and loads them into two registers L_1 and L_2, The registers are then updated as follows:

$$\begin{aligned} f_a(L_1) &= L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x_4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a \\ f_b(L_2) &= L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6]) \oplus k_b \\ L_1[i] &= L_1[i-1](i \le i \le |L_1|), L_1[0] = f_b(L_2), \\ L_2[i] &= L_2[i-1](i \le i \le |L_2|), L_2[0] = f_a(L_1), \end{aligned} \tag{2}$$

where $\oplus$ and $\cdot$ are bitwise XOR and AND operations respectively, $L[x]$ denotes the x-th bit of L, IR is the round constant value defined in the specification, and k_a and k_b are two subkey bits. For round i, k_a and k_b correspond to $sk_{2(i-1)}$ and $sk_{2(i-1)+1}$. The parameters of KATAN family are shown in Table 4. In this table, $|L_1|$ and $|L_2|$ denotes the lengths of registers L_1 and L_2, respectively.

The values of IR indicate whether irregular updates are used in the current round. If the value is 1, the irregular update rule will be applied during

Table 4: Parameters of the KATAN family

algorithm	$\|L_1\|$	$\|L_2\|$	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4	y_5	y_6
KATAN32	13	19	12	7	8	5	3	18	7	12	10	8	3
KATAN48	19	29	18	12	15	7	6	28	19	21	13	15	6
KATAN64	25	39	24	15	20	11	9	38	25	33	21	14	9

this round. If the value is 0, the irregular update rule will not be used. In KATAN48, functions f_a and f_b are used twice in a single round. The same subkeys are applied twice. Similarly for KATAN64, each round uses functions f_a and f_b three times with the same subkey bits.

2.4. SIMON *Specification*

The SIMON family of ciphers has a Feistel-like structure with AND as its nonlinear component [22]. Table 5 lists the specifications of all its variants SIMON-$2n/mn$ where n is the word size and m is the number of key words. SIMON's block size is $2n$ while its key length is mn. Since the key size is unrelated to our analysis, we will omit them in the naming convention used in the rest of the sections.

Table 5: SIMON specifications

Block Size($2n$)	**32**	**48**		**64**		**96**		**128**		
Key Size(mn)	64	72	96	96	128	96	144	128	192	256
Word Size(n)	16	24		32		48		64		
Key words(m)	4	3	4	3	4	2	3	2	3	4
Const Seq	z_0	z_0	z_1	z_2	z_3	z_2	z_3	z_2	z_3	z_4
Rounds T	32	36	36	42	44	52	54	68	69	72

SIMON's round function is mathematically defined as:

$$\begin{aligned} L_n &= R_{n-1} \oplus f(L_{n-1}) \oplus k_n \\ R_n &= L_{n-1}, \end{aligned} \tag{3}$$

where L and R represent the left and right inputs of the Feistel structure respectively, k_n represents the round key of the n-th round, and $f(x) = (ROTL_1(x) \wedge ROTL_8(x)) \oplus ROTL_2(x)$. The $ROTL_m$ function performs a circular left shift of m bits on its input. SIMON has a linear key schedule.

2.5. CHAM *Specification*

CHAM is a family of block ciphers that has a 4-branch generalized Feistel structure. Each specific cipher is noted as CHAM-n/k, where n represents

the block size in bits and k is the key size in bits. Given plaintext $P \in \{0,1\}^n$, ciphertext $C \in \{0,1\}^n$, and the key $K \in \{0,1\}^k$, the process of obtaining C involves applying r iterations of the round function on P with subkeys derived from K. The round function varies slightly based on the round number i. When $i \mod 2 = 1$, the round function is given by:

$$\begin{aligned} X_i &= X_i[0] \,||\, X_i[1] \,||\, X_i[2] \,||\, X_i[3] \\ X_i[3] &= ROTL_8((X_{i-1}[0] \oplus (i-1)) \\ &\quad \boxplus (ROTL_1(X_{i-1}[1]) \oplus RK[(i-1) \; mod \; 2k/w])) \\ X_i[j] &= X_{i-1}[j+1]. \end{aligned}$$

When $i \mod 2 = 0$, the round function is given by:

$$\begin{aligned} X_i &= X_i[0] \,||\, X_i[1] \,||\, X_i[2] \,||\, X_i[3] \\ X_i[3] &= ROTL_1((X_{i-1}[0] \oplus (i-1)) \\ &\quad \boxplus (ROTL_8(X_{i-1}[1]) \oplus RK[(i-1) \; mod \; 2k/w])) \\ X_i[j] &= X_{i-1}[j+1], \end{aligned}$$

where $X \boxplus Y$ denotes the addition of x and y modulo 2^w. For more details about the cipher, including its key scheduling algorithm, please refer to the cipher's original specifications [23].

Table 6: The parameters of CHAM ciphers

Cipher	n	k	r	w	$\frac{k}{w}$
CHAM-64/128	64	128	80	16	8
CHAM-128/128	128	128	80	32	4
CHAM-128/256	128	256	96	32	8

3. Boomerang Connectivity for AND-based Ciphers

3.1. Definition of $\wedge BCT$

In this section, we discuss the $\wedge$BCT, the bitwise AND ($\wedge$) counterpart of the BCT. Lightweight block ciphers such as KATAN and SIMON rely on the AND operator as their nonlinear component. Unlike ciphers with S-boxes where the differential probability depends on the number of active S-boxes, we calculate the differential probability for AND-based ciphers based on the

number of active difference bits. The condition for activating one difference bit in AND-based ciphers is defined as follows:

$$\Delta_x \vee \Delta_y = 1, \tag{4}$$

where Δ_x and Δ_y represent one-bit input differences to the AND operation. It is important to note that modular addition and S-boxes, which are typical choices for nonlinear components in many other block ciphers, are invertible. In contrast, the AND operator lacks invertibility, which implies that AND-based cryptographic primitives often have Feistel-like structures. Analyzing AND-based ciphers using boomerang attacks requires constructing a unique BCT tailored to the structure of the target cipher.

Next, we introduce our representation of an AND-based boomerang switch over a Feistel-like round. This representation will be used in our automatic boomerang search. The boomerang switch is shown in Figure 4 and is inspired by the FBCT[11]. We later show that the boomerang connectivity based on this structure can be generalised to different AND-based ciphers such as SIMON and KATAN. For easier definition of the $\wedge$BCT, we define a function A that takes a single (multi-bit) input difference, such as half of the plaintext, and returns the bits involved in an AND operation. The specific description of A will vary depending on the target cipher. For instance, in the case of the block cipher KATAN (see Section 6), A would return bits tapped from its registers.

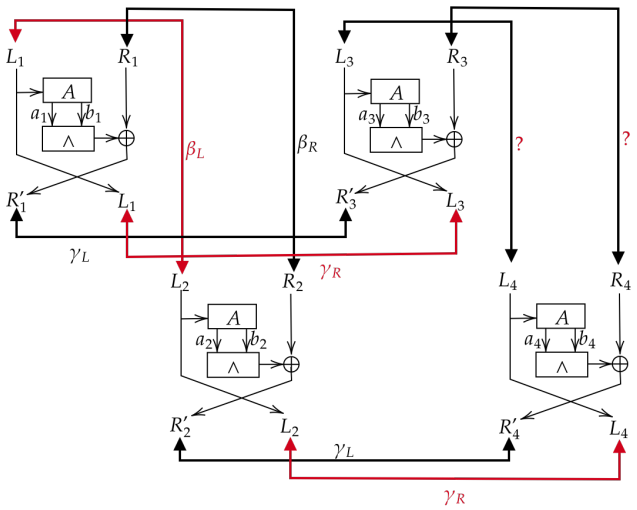

Figure 4: Boomerang switch over a Feistel-like round with an AND operation.

To ensure the validity of the boomerang switch, the difference between (L_3, L_4), and (R_3, R_4) must be equal to (L_1, L_2), and (R_1, R_2), respectively, as depicted in Figure 4. First, the difference β_L for L_3 and L_4 based on Figure 4 is calculated as:

$$\begin{aligned} L_3 \oplus L_4 &= (L_3 \oplus L_1) \oplus (L_1 \oplus L_2) \oplus (L_2 \oplus L_4) \\ &= \gamma_R \oplus \beta_L \oplus \gamma_R = \beta_L. \end{aligned} \tag{5}$$

In any scenario, the difference between L_3 and L_4 is always equal to the difference between L_1 and L_2. Equation 6 shows that the difference between R_3 and R_4 is only related to L_1, β_L and γ_R:

$$\begin{aligned} R_3 \oplus R_4 &= (\wedge_F(A(L_1)) \oplus R_1 \oplus \gamma_L \oplus \wedge_F(A(L_1 \oplus \gamma_R))) \\ &\quad \oplus(\wedge_F(A(L_1 \oplus \beta_L \oplus \gamma_R)) \oplus R_1 \oplus \beta_R \oplus \\ &\quad \wedge_F(A(L_1 \oplus \beta_L)) \oplus \gamma_L) \\ &= \wedge_F(A(L_1)) \oplus \wedge_F(A(L_1 \oplus \gamma_R)) \oplus \\ &\quad \wedge_F(A(L_1 \oplus \beta_L)) \oplus \wedge_F(A(L_1 \oplus \beta_L \oplus \gamma_R)) \oplus \beta_R. \end{aligned} \tag{6}$$

Note that $\wedge_F$ represents a function that performs a bitwise AND on the two values returned by A. To ensure that the difference $R_3 \oplus R_4$ is consistent with $R_1 \oplus R_2$, the following condition must be met:

$$\begin{aligned} \wedge_F(A(L_1)) \oplus \wedge_F(A(L_1 \oplus \gamma_R)) \oplus \wedge_F(A(L_1 \oplus \beta_L)) \oplus \\ \wedge_F(A(L_1 \oplus \beta_L \oplus \gamma_R)) = 0. \end{aligned} \tag{7}$$

With these requirements and notations in place, we now define ∧BCT:

Definition 3. *(1-bit ∧BCT). Let a and b be two 1-bit outputs from the function A that takes L_1 as an input. Let the bit indexes of a and b in L_1 be m and n, respectively. We denote β_L^x and γ_R^y as the bits of β_L and γ_R located at indexes x and y, respectively. Then, Equation 7 can be rewritten as:*

$$\begin{aligned}(a \wedge b) \oplus ((a \oplus \gamma^m) \wedge (b \oplus \gamma^n)) \oplus ((a \oplus \beta^m) \wedge \\ (b \oplus \beta^n)) \oplus ((a \oplus \beta^m \oplus \gamma^m) \wedge (b \oplus \beta^n \oplus \gamma^n)) = 0.\end{aligned} \tag{8}$$

Then, a 1-bit ∧BCT is given by:

$$\begin{aligned}\wedge BCT(\beta^m, \beta^n, \gamma^m, \gamma^n) \quad = \quad & \#\{a, b \in \{0,1\} | a \wedge b \oplus \\ & (a \oplus \beta^m) \wedge (b \oplus \beta^n) \oplus \\ & (a \oplus \gamma^m) \wedge (b \oplus \gamma^n) \\ & \oplus (a \oplus \beta^m \oplus \gamma^m) \\ & \wedge (b \oplus \beta^n \oplus \gamma^n) = 0\}.\end{aligned} \tag{9}$$

The probability of the boomerang switch for a 1-bit AND is expressed as:

$$P_{\wedge BCT}(\beta^m, \beta^n, \gamma^m, \gamma^n) = \frac{\wedge BCT(\beta^m, \beta^n, \gamma^m, \gamma^n)}{4}. \tag{10}$$

The switching effect for each bit is evaluated independently. Therefore, the overall switching probability P_{switch} for AND operations with i-bit operands is the product of the switching probabilities of each bit:

$$P_{switch} = \prod_{j=0}^{i-1} P_{\wedge BCT_j}, \tag{11}$$

where $P_{\wedge BCT_j}$ is the switching probability for the j-th bit of the AND operation.

Property 1. *(Constraints for a 1-bit valid boomerang switch). Taking into account the computational properties of $GF(2)$ and Equation 8, the constraint for valid 1-bit boomerang switches in AND-based ciphers can be simplified to:*

$$
\begin{aligned}
&(a \wedge b) \oplus ((a \oplus \gamma^m) \wedge (b \oplus \gamma^n)) \oplus ((a \oplus \beta^m) \wedge \\
&(b \oplus \beta^n)) \oplus ((a \oplus \beta^m \oplus \gamma^m) \wedge (b \oplus \beta^n \oplus \gamma^n)) \\
=&(a \wedge b) \oplus ((a \wedge b \oplus a \wedge \gamma^n \oplus \gamma^m \wedge b \oplus \gamma^m \wedge \gamma^n)) \oplus \\
&((a \wedge b \oplus a \wedge \beta^n \oplus \beta^m \wedge b \oplus \beta^m \wedge \beta^n)) \oplus \\
&((a \wedge b \oplus a \wedge \beta^n \oplus a \wedge \gamma^n \oplus \beta^m \wedge b \oplus \beta^m \wedge \beta^n \oplus \beta^m \wedge \gamma^n \oplus \\
&\gamma^m \wedge b \oplus \gamma^m \wedge \beta^n \oplus \gamma^m \wedge \gamma^n)) \\
=&(a \wedge b) \oplus (a \wedge b) \oplus (a \wedge \gamma^n) \oplus (\gamma^m \wedge b) \oplus (\gamma^m \wedge \gamma^n) \oplus \\
&(a \wedge b) \oplus (a \wedge \beta^n) \oplus (\beta^m \wedge b) \oplus (\beta^m \wedge \beta^n) \oplus (a \wedge b) \oplus \\
&(a \wedge \beta^n) \oplus (a \wedge \gamma^n) \oplus (\beta^m \wedge b) \oplus (\beta^m \wedge \beta^n) \oplus (\beta^m \wedge \gamma^n) \oplus \\
&(\gamma^m \wedge b) \oplus (\gamma^m \wedge \beta^n) \oplus (\gamma^m \wedge \gamma^n) \\
=&(\beta^m \wedge \gamma^n) \oplus (\gamma^m \wedge \beta^n) = 0
\end{aligned}
$$

Based on the constraints of a valid 1-bit boomerang switch in Property 1, a 1-bit valid boomerang switch must fulfil the following equality:

$$\beta^m \wedge \gamma^n = \beta^n \wedge \gamma^m \tag{12}$$

This simplified representation is used in the automated boomerang search model described in Section 5 to determine the validity of boomerang switches, rather than relying on the complete boomerang table.

3.2. Properties of ∧BCT

Table 7: The 1-bit $\wedge BCT$

$\beta^m \|\| \beta^n$ \ $\gamma^m \|\| \gamma^n$	00	01	10	11
00	4	4	4	4
01	4	4	0	0
10	4	0	4	0
11	4	0	0	4

In this section, we look into the various properties of the ∧BCT. Table 7 illustrates how the 1-bit ∧BCT maps two independent input difference bits into a single output difference bit. From this table, we can observe that the

switching probability is either deterministic or impossible. The following are properties of the ∧BCT:

Ladder switch: The entries in the first row and first column of ∧BCT is equal to 4 (the boomerang switch holds with probability 1). This property is similarly observed in other BCT variants and has been described in [24].

Feistel switch: When considering any pairs of differences with $\beta^m = \beta^n = \gamma^m = \gamma^n$, the corresponding entry in the boomerang switch table also has a value of 4, which implies that the switch is deterministic.

4. Boomerang Connectivity for Modular Addition

4.1. Definition of ABCT

In this section, we introduce the generalized form of the BCT for ADD (ABCT). We will recap its properties used to find rectangle distinguishers for CHAM.

Definition 4. *((Generalized) ABCT). Let $\alpha, \alpha', \beta, \beta' \in \mathbb{F}_2^n$. The generalized Boomerang Connectivity Table for addition modulo 2^n involving two n-bit addends can be described by a four-dimensional table, in which the entry for $(\alpha, \alpha', \beta, \beta')$ is computed as:*

$$\begin{aligned} ABCT_{\boxplus}(\alpha, \alpha', \beta, \beta') \;=\; & \#\{(x, x') \in \mathbb{F}_2^n | ((x \boxplus x') \oplus \beta) \\ & \boxminus (x' \oplus \beta')) \oplus ((((x \oplus \alpha) \boxplus \\ & (x' \oplus \alpha')) \oplus \beta) \boxminus (x' \oplus \alpha' \\ & \oplus \beta')) = \alpha\}. \end{aligned}$$

The ABCT originally described by Cid et al. [8] can be derived from this definition by having $\beta' = 0$, which fixes one of the addends on opposing faces of the boomerang. When it comes to actual ARX ciphers, all differences involved in an ADD (or SUB) operation may not be zero. As such, the generalized ABCT can more accurately describe the boomerang switch over one round for ARX ciphers. Moving forward in the rest of the paper, we will refer to this generalized version simply as ABCT. This definition of the ABCT was used by Wang et al. [16] in their automated boomerang search tool applied to SPECK and LEA.

4.2. Properties of ABCT

In all of the following scenarios, the boomerang switch occurs with probability 1.

Ladder switch: Occurs when either both addend differences specified by the upper trail (α,α') or the output differences specified by the lower trail (β,β') are zero.

Most significant bit (MSB) switch: Occurs when the only active (difference) bit in either β or β' is the MSB while the other has a zero difference (β, β') =(`0b100...0,0b000...0`) or the MSB is the only active bit in both (β, β') =(`0b100...0,0b100...0`). The MSB switch also occurs when the only active bit is the MSB in either α or α' while the other has a zero difference, or when the MSB is active in both.

5. An Automated Search for Boomerang-style Distinguishers for AND-RX Ciphers

5.1. Search Strategy

In this section, we propose an automated search for boomerang and rectangle distinguishers for AND-RX ciphers based on SMT solvers. More specifically, we extend the functionalities of CryptoSMT [25] to automatically search for boomerang distinguishers and compute their corresponding probabilities. One main advantage of the proposed tool is its ability to construct rectangle distinguishers by considering all possible trails in E_0 that start with a difference α and all possible trails in E_1 that end with δ, as long as their switch in E_m is valid according to $\wedge$BCT. We apply the automatic search to KATAN and SIMON, assuming round independence within each cipher.

The search strategy comprises two main steps. First, we search for the best single boomerang characteristic starting from an input difference α to an output difference δ with a valid switch from β to γ. The overall weight of the initial boomerang trail is denoted as w. Next, we fix α and δ and continue to search for all possible trails. The SMT model will sum all probabilities 2^{-w} of the valid trails, then return the final rectangle probability, $2^{-w'}$. In the previous notation, w refers to the *weight* of a trail or distinguisher. The

relationship between the weight of a single boomerang trail (w), the weight of a rectangle (w'), and the actual weight (W) is:

$$2^{-w} < 2^{-w'} \approx 2^{-W}. \tag{13}$$

Upon completing the rectangle search, the probability is computed using the following formula:

$$P_{rect} = \sum_{i=1}^{n} (2^{-w_i} \cdot SOL_i), \tag{14}$$

where SOL_i denotes the number of boomerang trails with weight w_i while n represents the number of unique boomerang trails found.

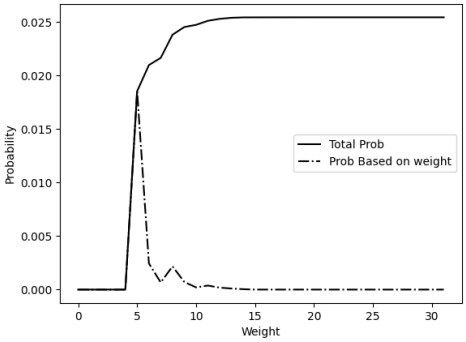

Figure 5: Probabilistic trends in the rectangle search.

The proposed approach leverages the differential effect by *clustering* all valid boomerangs trails that start from α and end with δ. In practical scenarios, we cannot exhaustively search for all possible boomerangs to construct a rectangle distinguisher due to time constraints. Therefore, we introduce a threshold variable that assists in deciding when to stop the search. Figure 5 illustrates the variation in distinguishing probability during the rectangle search. The solid line represents the total probability of the rectangle distinguisher after including trails of a specific weight while the dotted line represents the total probability of all trails with a given weight. As the individual boomerang trail weights increase, the rectangle probability will eventually converge to a certain value.

When constructing a rectangle distinguisher, we search for boomerang trails starting from a lower weight and iteratively increase the weight once no more boomerang trails can be found. We set a threshold to limit the number of attempts to find new boomerang trails. For instance, if the threshold is set to 5 and the current weight of the rectangle distinguisher is $w' = 13.3224$, the search stops if the integer value of $\lfloor w' \rfloor$ remains unchanged after five new boomerang trails have been included in the rectangle distinguisher. Since solving SAT/SMT problems is NP-complete, running for a longer duration can yield better results. Utilizing the threshold variable allows us to effectively determine when to stop the rectangle search.

5.2. Verification of Correctness

Verification of a boomerang or rectangle distinguisher's *actual* distinguishing probability presents a notable challenge. For instance, a cipher with a 32-bit block size requires exhaustive testing of 2^{31} ciphertext pairs to verify a boomerang distinguisher, consuming a substantial amount of time. The entire process involves the encryption of each plaintext pair corresponding to the input difference ($P_1 \oplus P_2 = \alpha$) of the boomerang (or rectangle) in one direction, computing their corresponding ciphertext pairs based on the output difference ($C_3 = C_1 \oplus \delta, C_4 = C_2 \oplus \delta$), then decrypting them to check if $P_3 \oplus P_4 = \alpha$. To make this process more efficient, we devised a CUDA program to verify boomerang distinguishers for 32-bit block ciphers. Rather than taking 42 hours to complete on an Intel i7-9750H CPU, verification now requires a mere 20 minutes on an Nvidia GTX 1660TI graphics card.

Our experiments on 32-bit KATAN and SIMON show that the theoretically derived rectangle distinguishers found using the proposed SMT search generally correspond to actual ones in practice. As the theoretical boomerang probability is an average over all keys under the assumption that all key bits are independent, there are a few instances whereby the probabilities are either higher or lower than estimated by the SMT model. All experimental results are discussed in Section 6 and Section 7.

6. Application to KATAN

6.1. Boomerang Properties of the KATAN *Family*

There are four AND operators in KATAN's round function, each independent of one another. Since one of the AND operators involves a fixed IR bit, input differences will propagate deterministically. We only need to

consider the remaining three AND operators as the focus of our analysis. We will construct $\wedge BCT$ constraints for the L_1 and L_2 registers based on Equation 12.

Recall that (m, n) are indexes of bits involved in an AND operation. Taking KATAN32 as an example, we have $(m, n) = (8, 5)$ for L_1 while for L_2 the indexes involved in two AND operations are denoted as $(m', n') = (12, 10)$ and $(m'', n'') = (8, 3)$ respectively. We can derive the boomerang switching constraints for L_1 directly from Equation 12 as:

$$L_1 : \beta^m \wedge \gamma^{n+1} = \beta^n \wedge \gamma^{m+1}, \tag{15}$$

where β and γ are the input and output differences of the boomerang switch respectively. Note that the bit indexes for γ are shifted by 1 bit due to the clocking of the shift registers. Since the results of the two AND operations in L_2 will be XOR-ed, we can construct the following $\wedge BCT$ constraints:

$$\begin{aligned} L_2 : (\beta^{m'} \wedge \gamma^{n'+1}) \oplus (\beta^{m''} \wedge \gamma^{n''+1}) = (\beta^{n'} \wedge \gamma^{m'+1}) \oplus (\beta^{n''} \wedge \gamma^{m''+1}), \\ (\beta^{m'} \wedge \gamma^{n'+1}) \oplus (\beta^{n'} \wedge \gamma^{m'+1}) = (\beta^{m''} \wedge \gamma^{n''+1}) \oplus (\beta^{n''} \wedge \gamma^{m''+1}). \end{aligned} \tag{16}$$

Based on Table 4, the indexes involved in the $\wedge BCT$ constraints are $(m, n, m', n', m'', n'') = (x_3, x_4, y_3, y_4, y_5, y_6)$.

For all variants of KATAN, only certain input bits are involved in each encryption round while the rest are merely shifted by 1 bit. As such, we can derive a trivial multi-round boomerang switch for KATAN. We analyzed the three variants of KATAN and derived a formula for computing the maximum number of independent rounds for the aforementioned multi-round switch.

Property 2. *Let Z represent the number of times* KATAN*'s round function is applied (or the number of times the registers are clocked) in one encryption round and y_6 is the register bit index taken from the* KATAN *parameter table (Table 4). The maximum number of independent rounds is then given by*

$$r_m = \lfloor \frac{y_6 + 1}{Z} \rfloor. \tag{17}$$

Proof. The length of the multi-round switch depends on the minimum number of rounds required for the output of one nonlinear operation to serve as the input for another nonlinear operation. Rounds in which the nonlinear outputs are simply shifted are referred to as *independent* rounds. There are two nonlinear output bits computed each round, one for each register. When

the registers are clocked, these nonlinear output bits are shifted into the least significant bit of the other register, i.e. from L_1 to L_2 and vice versa. The first two bits involved in AND operations have indexes x_5 and y_6 in L_1 and L_2 respectively. It also happens that $x_5 = y_6$ for all KATAN variants. Since x_5 is involved in an AND operation with the IR bit, its difference will propagate deterministically. Hence, the number of independent rounds is based on y_6. We will need to clock the registers at least $y_6 + 1$ times before the nonlinear bit coming from L_1 will be involved in another AND operation. Taking KATAN32 as an example where $y_6 = 3$, the registers need to be clocked at least 4 times:

1. $L_2[0] = f_a(L_1)$
2. $L_2[1] = L_2[0]$
3. $L_2[2] = L_2[1]$
4. $L_2[3] = L_2[2]$

The same property can be observed for KATAN48 and KATAN64 where $y_6 = 6$ and $y_6 = 9$ respectively. Since the different KATAN variants clock their registers Z times per encryption round, the maximum number of independent rounds is $\lfloor \frac{y_6+1}{Z} \rfloor$. □

For KATAN32, $Z = 1$; for KATAN48, $Z = 2$; for KATAN64, $Z = 3$. Using this formula, the maximum number of independent rounds for the KATAN family of ciphers is 4 rounds for KATAN32, 3 rounds for KATAN48, and 3 rounds for KATAN64. These independent rounds will be encoded into the proposed automated boomerang search.

6.2. Single-key Rectangle Distinguishers for KATAN*32*

Based on Equation 15 and Equation 16, we can use the following constraints to construct a 4-round boomerang switch for KATAN32:

$$L_1 = \begin{cases} \beta^m \wedge \gamma^{n+1} = \beta^n \wedge \gamma^{m+1}, & \text{(L1.1)} \\ \beta^{m-1} \wedge \gamma^n = \beta^{n-1} \wedge \gamma^m, & \text{(L1.2)} \\ \beta^{m-2} \wedge \gamma^{n-1} = \beta^{n-2} \wedge \gamma^{m-1}, & \text{(L1.3)} \\ \beta^{m-3} \wedge \gamma^{n-2} = \beta^{n-3} \wedge \gamma^{m-2}, & \text{(L1.4)} \end{cases}$$

$$L_2 = \begin{cases} (\beta^{m'} \wedge \gamma^{n'+1}) \oplus (\beta^{n'} \wedge \gamma^{m'+1}) = (\beta^{m''} \wedge \gamma^{n''+1}) \oplus (\beta^{n''} \wedge \gamma^{m''+1}), & \text{(L2.1)} \\ (\beta^{m'-1} \wedge \gamma^{n'}) \oplus (\beta^{n'-1} \wedge \gamma^{m'}) = (\beta^{m''-1} \wedge \gamma^{n''}) \oplus (\beta^{n''-1} \wedge \gamma^{m''}), & \text{(L2.2)} \\ (\beta^{m'-2} \wedge \gamma^{n'-1}) \oplus (\beta^{n'-2} \wedge \gamma^{m'-1}) = (\beta^{m''-2} \wedge \gamma^{n''-1}) \oplus (\beta^{n''-2} \wedge \gamma^{m''-1}), & \text{(L2.3)} \\ (\beta^{m'-3} \wedge \gamma^{n'-2}) \oplus (\beta^{n'-3} \wedge \gamma^{m'-2}) = (\beta^{m''-3} \wedge \gamma^{n''-2}) \oplus (\beta^{n''-3} \wedge \gamma^{m''-2}), & \text{(L2.4)} \end{cases}$$

where β is the input difference to the first round of the boomerang switch, and γ is the output difference after the fourth round. The L1.1 constraint is taken directly from Equation 15. The bits involved in L1.2 to L1.4 are those that will be shifted into position (m, n) after clocking the L_1 register each round. Similarly, the L2.1 constraint is taken directly from Equation 16 while the rest (L2.2 to L2.4) involve bits that will be shifted into positions (m', n') and (m'', n'') after clocking the L_2 register each round.

We implement these constraints to search for boomerang and rectangle distinguishers for KATAN32 in the single-key setting with a search threshold of 5 (the same threshold value is used for all KATAN variants), the results of which are as follows:

Improved 83-round Rectangle Distinguisher for KATAN32: The longest single-key rectangle distinguisher currently published in the literature is 83 rounds with a probability of $2^{-21.78}$ [18]. However, this rectangle distinguisher was based on the previous (and now proven to be inaccurate [20]) assumption that two independent E_0 and E_1 would be compatible. We recompute the probability of this rectangle distinguisher (α = (`0000,8081`), δ = (`0080,1081`)) using the proposed search strategy in Section 5. With the incorporation of the boomerang switch, our SMT model returned a boomerang probability of 2^{-100} and a rectangle probability of $2^{-74.24}$, indicating that this trail is not a valid distinguisher.

We then searched for an improved 83-round rectangle distinguisher. Using the proposed search strategy, we found an 83-round rectangle discriminator with a probability of 2^{-48} and a rectangle probability of $2^{-28.65}$, the GPU checker gives a probability of $2^{-26.7}$. The round distribution of the distinguisher (E_0, E_m, E_1) was $(r_0, r_m, r_1) = (39, 4, 40)$. The input and output differences are $\alpha =$ (`0000,8801`) and $\delta =$ (`0010,0210`), respectively.

86-round Rectangle Distinguisher for KATAN32: Our next goal was to find the longest possible single-key rectangle distinguisher for KATAN32. We were able to find an 86-round boomerang distinguisher that holds with a probability of 2^{-44} where $(r_0, r_m, r_1) = (42, 4, 42)$. Based on its input and output differences of $\alpha =$ (`1006,8880`) and $\delta =$ (`00D0,1081`), we computed its rectangle probability as $2^{-30.06}$, the GPU checker returned a probability of $2^{-28.06}$. We also found the best 84 and 85-round rectangle distinguishers as summarised in Table 1.

6.3. Single-key Rectangle Distinguishers for KATAN*48*

We first derive the constraints for constructing a 3-round boomerang switch for KATAN48, similar to how we did for KATAN32. Since KATAN48 clocks each register twice each round, L_1 and L_2 will each have 6 sets of constraints to represent the 3-round switch. The following are constraints derived based on Equation 15 and Equation 16 using indexes taken from Table 4, where $(m, n, m', n', m'', n'') = (15, 7, 21, 13, 15, 6)$:

$$L_1 = \begin{cases} \beta^{15} \wedge \gamma^{8} = \beta^{7} \wedge \gamma^{16}, \ \beta^{14} \wedge \gamma^{7} = \beta^{6} \wedge \gamma^{15} \\ \beta^{13} \wedge \gamma^{6} = \beta^{5} \wedge \gamma^{14}, \ \beta^{12} \wedge \gamma^{5} = \beta^{4} \wedge \gamma^{13} \\ \beta^{11} \wedge \gamma^{4} = \beta^{3} \wedge \gamma^{12}, \ \beta^{10} \wedge \gamma^{3} = \beta^{2} \wedge \gamma^{11} \end{cases}$$

$$L_2 = \begin{cases} (\beta^{21} \wedge \gamma^{14}) \oplus (\beta^{13} \wedge \gamma^{22}) = (\beta^{15} \wedge \gamma^{7}) \oplus (\beta^{6} \wedge \gamma^{16}) \\ \qquad (\beta^{20} \wedge \gamma^{13}) \oplus (\beta^{12} \wedge \gamma^{21}) = (\beta^{14} \wedge \gamma^{6}) \oplus (\beta^{5} \wedge \gamma^{15}) \\ (\beta^{19} \wedge \gamma^{12}) \oplus (\beta^{11} \wedge \gamma^{20}) = (\beta^{13} \wedge \gamma^{5}) \oplus (\beta^{4} \wedge \gamma^{14}) \\ \qquad (\beta^{18} \wedge \gamma^{11}) \oplus (\beta^{10} \wedge \gamma^{19}) = (\beta^{12} \wedge \gamma^{4}) \oplus (\beta^{3} \wedge \gamma^{13}) \\ (\beta^{17} \wedge \gamma^{10}) \oplus (\beta^{9} \wedge \gamma^{18}) = (\beta^{11} \wedge \gamma^{3}) \oplus (\beta^{2} \wedge \gamma^{12}) \\ \qquad (\beta^{16} \wedge \gamma^{9}) \oplus (\beta^{8} \wedge \gamma^{17}) = (\beta^{10} \wedge \gamma^{2}) \oplus (\beta^{1} \wedge \gamma^{11}) \end{cases}$$

Improved 60-round Rectangle Distinguisher for KATAN48: In current literature, the longest single-key boomerang distinguisher identified for KATAN48 has 60 rounds with differences $\alpha =$ (`0000,0090,4000`) and $\delta =$ (`0004,0200,0000`), and a claimed rectangle probability of $2^{-23.36}$ [18]. We re-evaluated the rectangle probability using our search tool and found it to be higher than originally estimated ($2^{-21.52}$).

Next, we searched for an improved 60-round rectangle distinguisher. We first found a boomerang distinguisher with input and output differences of $\alpha =$ (`0000,0402,0000`) and $\delta =$ (`0002,0100,0000`), respectively that holds with a probability of 2^{-38}. The round distribution was $(r_0, r_m, r_1) = (29, 3, 28)$. Based on these settings the corresponding rectangle probability was determined to be $2^{-17.51}$.

83-round Rectangle Distinguisher for KATAN48: The longest distinguisher for KATAN48 that we could find has 83 rounds, with a round distribution of $(r_0, r_m, r_1) = (40, 3, 40)$. The input and output differences are $\alpha =$ (`0000,0090,4000`) and $\delta =$ (`0090,4000,000D`). The boomerang and rectangle probabilities for the distinguisher were 2^{-66} and $2^{-41.27}$ respectively. We also found the best 81 and 82-round single-key rectangle distinguishers to date, both of which are depicted in Table 1.

6.4. Single-key Rectangle Distinguishers for KATAN*64*

Since KATAN64 clocks each register three times each round, L_1 and L_2 will each have 9 sets of constraints to represent the 3-round switch. The following are constraints derived based on Equation 15 and Equation 16 using indexes taken from Table 4, where $(m, n, m', n', m'', n'') = (20, 11, 33, 21, 14, 9)$:

$$L_1 = \begin{cases} \beta^{20} \wedge \gamma^{12} = \beta^{11} \wedge \gamma^{21}, & \beta^{19} \wedge \gamma^{11} = \beta^{10} \wedge \gamma^{20}, \\ & \beta^{18} \wedge \gamma^{10} = \beta^{9} \wedge \gamma^{19} \\ \beta^{17} \wedge \gamma^{9} = \beta^{8} \wedge \gamma^{18}, & \beta^{16} \wedge \gamma^{8} = \beta^{7} \wedge \gamma^{17}, \\ & \beta^{15} \wedge \gamma^{7} = \beta^{6} \wedge \gamma^{16} \\ \beta^{14} \wedge \gamma^{6} = \beta^{5} \wedge \gamma^{15}, & \beta^{13} \wedge \gamma^{5} = \beta^{4} \wedge \gamma^{14}, \\ & \beta^{12} \wedge \gamma^{4} = \beta^{3} \wedge \gamma^{13} \end{cases}$$

$$L_2 = \begin{cases} (\beta^{33} \wedge \gamma^{22}) \oplus (\beta^{21} \wedge \gamma^{34}) = (\beta^{15} \wedge \gamma^{10}) \oplus (\beta^{9} \wedge \gamma^{15}) \\ \quad (\beta^{32} \wedge \gamma^{21}) \oplus (\beta^{20} \wedge \gamma^{33}) = (\beta^{14} \wedge \gamma^{9}) \oplus (\beta^{8} \wedge \gamma^{14}) \\ \quad\quad (\beta^{31} \wedge \gamma^{20}) \oplus (\beta^{19} \wedge \gamma^{32}) = (\beta^{13} \wedge \gamma^{8}) \oplus (\beta^{7} \wedge \gamma^{13}) \\ (\beta^{30} \wedge \gamma^{19}) \oplus (\beta^{18} \wedge \gamma^{31}) = (\beta^{12} \wedge \gamma^{7}) \oplus (\beta^{6} \wedge \gamma^{12}) \\ \quad (\beta^{29} \wedge \gamma^{18}) \oplus (\beta^{17} \wedge \gamma^{30}) = (\beta^{11} \wedge \gamma^{6}) \oplus (\beta^{5} \wedge \gamma^{11}) \\ \quad\quad (\beta^{28} \wedge \gamma^{17}) \oplus (\beta^{16} \wedge \gamma^{29}) = (\beta^{10} \wedge \gamma^{5}) \oplus (\beta^{4} \wedge \gamma^{10}) \\ (\beta^{27} \wedge \gamma^{16}) \oplus (\beta^{15} \wedge \gamma^{28}) = (\beta^{9} \wedge \gamma^{4}) \oplus (\beta^{3} \wedge \gamma^{9}) \\ \quad (\beta^{26} \wedge \gamma^{15}) \oplus (\beta^{14} \wedge \gamma^{27}) = (\beta^{8} \wedge \gamma^{3}) \oplus (\beta^{2} \wedge \gamma^{8}) \\ \quad\quad (\beta^{25} \wedge \gamma^{14}) \oplus (\beta^{13} \wedge \gamma^{26}) = (\beta^{7} \wedge \gamma^{2}) \oplus (\beta^{1} \wedge \gamma^{7}) \end{cases}$$

Improved 56-round Rectangle Distinguisher for KATAN64: In current literature, the longest single-key rectangle distinguisher for KATAN64 had 56 rounds, with a probability of $2^{-44.26}$. Its input and output differences were $\alpha =$ (`0000,0000,0400,2001`) and $\delta =$ (`0020,1100,8000,0000`) respectively [18]. Upon verification, we found that the boomerang and rectangle probabilities for this distinguisher were instead 2^{-88} and $2^{-67.54}$ respectively. We instead found a different boomerang distinguisher with differences $\alpha =$ (`0000,0010,0080,0400`) and $\delta =$ (`0020,3100,8000,0000`). The boomerang probability was 2^{-56} which was improved to $2^{-51.92}$ when considering the rectangle framework. The number of rounds for (E_0, E_m, E_1) were $(r_0, r_m, r_1) = (27, 3, 26)$.

62-round Rectangle Distinguisher for KATAN64: Our search for the longest distinguisher uncovered a 62-round boomerang that holds with probability 2^{-68}. Its input and output differences were $\alpha = (\texttt{0000,0020,0500,2001})$ and $\delta = (\texttt{2011,0080,0000,0040})$ respectively, with $(r_0, r_m, r_1) = (30, 3, 29)$. When considering all other boomerang trails and switches, the resulting rectangle probability was $2^{-62.25}$. Table 1 summarizes our findings, including 60 and 61-round rectangles for KATAN64.

7. Application to SIMON

7.1. Prior Differential Attacks on SIMON

To the best of our knowledge, no boomerang attacks on SIMON have been previously reported in the single-key setting while findings in the related-key setting have been recently published by Bonnetain and Lallemand [17]. The most best single-key (non-boomerang) differential attack on SIMON32 was based on a 13-round differential with a probability of $2^{-30.22}$. The input and output differences of this differential were $\alpha = (\texttt{0000,0040})$ and $\delta = (\texttt{4000,0000})$ respectively. For SIMON48, the best single-key (non-boomerang) differential attack utilized a 15-round differential with a probability of $2^{-43.01}$. The input and output differences of this differential were $\alpha = (\texttt{0101,0004,4040})$ and $\delta = (\texttt{4440,4010,0000})$ respectively.

Finally, the best single-key (non-boomerang) differential attack on SIMON64 relied on a 21-round differential that had a probability of $2^{-61.01}$. The input and output differences for this differential were $\alpha = (0000, 0100, 0000, 0440)$ and $\delta = (0000, 0440, 0000, 0100)$ [19].

Table 8: Bit indexes involved in the AND operation for SIMON32

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$m = ROTL_1(x)$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
$n = ROTL_8(x)$	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7

7.2. Single-key Boomerang Distinguishers for SIMON

For SIMON-n, there are a total of $\frac{n}{2}$ single-bit AND operations. In Table 8, we denote x as the original indexes of bits involved in the AND operation while the second and third rows of each column represent the same bit indexes after bitwise rotations have been applied. Consequently, there are 16 switch constraints, each following the general form:

$$\beta^m \wedge \gamma^n = \beta^n \wedge \gamma^m, \tag{18}$$

where m and n are bit indexes shown in Table 8. These constraints were incorporated in our automated search for rectangle distinguishers with a search threshold of 10. The results of the boomerang search are as follows:

13-round Rectangle Distinguisher for SIMON32: In the single-key setting, we successfully identified a boomerang trail with a probability of 2^{-44}, where $(r_0, r_m, r_1) = (6, 1, 6)$. Based on its input and output differences of $\alpha = (\texttt{0010,0044})$ and $\delta = (\texttt{1100,0400})$ respectively, its corresponding rectangle probability was $2^{-28.54}$. The GPU checker returned the probability as $2^{-28.04}$.

15-round Rectangle Distinguisher for SIMON48: While searching for the longest rectangle distinguisher for SIMON48, we found a 15-round rectangle distinguisher that has a probability of $2^{-41.87}$, with $\alpha = (0000, 4000, 4111)$ and $\delta = (0006, 4000, 0100)$. The 15 rounds were divided into $(r_0, r_m, r_1) = (7, 1, 7)$.

16-round Rectangle Distinguisher for SIMON48: The maximum number of rounds for a valid single-key rectangle distinguisher for SIMON48 was 16. It has a probability of $2^{-46.45}$ with input and output differences $\alpha = (\texttt{0400,0019,1000})and\delta = (\texttt{2000,8280,0000})$ respectively. The 16 rounds were divided into $(r_0, r_m, r_1) = (7, 1, 8)$. A summary of all SIMON boomerang and rectangle distinguishers found using the proposed tool is listed in Table 2.

8. Application to CHAM

8.1. Single-key Rectangle Distinguishers for CHAM

In this section, we first revisit the best boomerang distinguisher found so far – the one proposed by the designers of CHAM. We then introduce new rectangle distinguishers for CHAM based on a 1-round boomerang switch. In each round, only two out of four words are involved in the ADD operation. Therefore, by restricting E_m to just one round, bits that are *not* involved in the ADD operation can take on any value during the boomerang switch. This can be viewed as a truncated boomerang switch that allows us to cluster more boomerang trails to maximize distinguishing probability. The boomerang switches were analyzed using ABCT and verified experimentally. For differential search tasks, we use CryptoSMT [25].

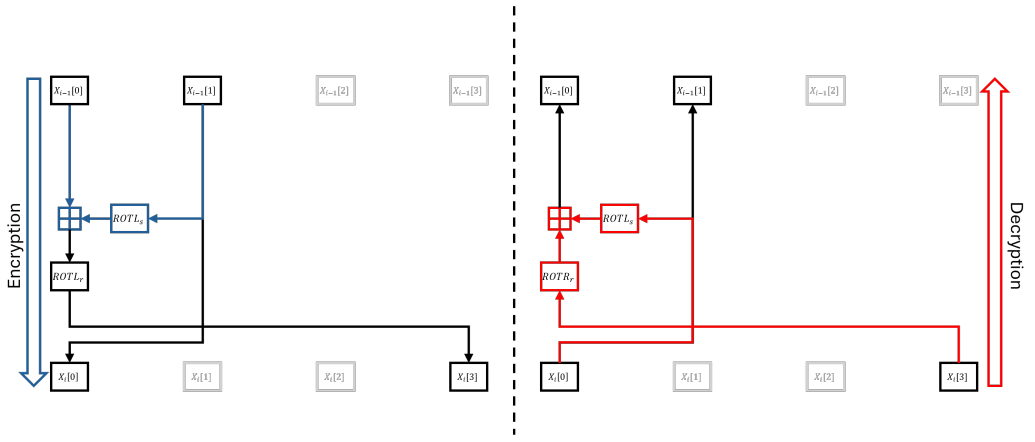

Figure 6: Evaluation of CHAM's boomerang switch

In the following descriptions, $ROTR_m$ and $ROTL_m$ functions perform m-bit circular right and left shifts of their inputs respectively. Only two of CHAM's four branches are involved in ADD and SUB operations in the forward (encryption) and backward (decryption) directions respectively. Since the ABCT is constructed based on values directly involved in ADD or SUB operations, we will need to pre-rotate some of the branches as shown in Figure 6. In the encryption direction, $X_{i-1}[1]$ needs to undergo a right rotation while in the decryption direction, $X_i[0]$ and $X_i[3]$ need to undergo a left and right rotation respectively before they are used as inputs to the ABCT.

36-round rectangle distinguisher: So far, the longest boomerang distinguisher was the one proposed by the designers of CHAM. They combined a 17-round E_0 trail (8200,0100,0001,8000) $\rightarrow$ (0400,0004,0502,0088) with an 18-round E_1 trail (8200,0100,0001,8000) $\rightarrow$ (0004,0502,0088,0000) to form a 35-round boomerang [23]. No switching round was included in their analysis. While verifying the validity of the trails, we found that the proposed 18-round E_1 trail started from an odd-numbered round. Since E_0 itself has an odd number of rounds, E_1 should start on an even-numbered round which has a different set of rotation values. To fix this error, we bridge the gap between E_0 and E_1 by adding 1 round of E_m. This then allows the use of the trails found by the designers. By considering the differential effect, E_0 and E_1 have improved differential probabilities of $2^{-15} \rightarrow 2^{-14.35}$ and $2^{-16} \rightarrow 2^{-15.13}$ respectively. The switching effect occurs with probability

$ABCT(\texttt{0400}, ROTL_8(\texttt{0004}), ROTR_1(\texttt{8000}), ROTL_8(\texttt{8200}))/2^{32} \approx 2^{31.71-32} = 2^{-0.29}$. Experimentally, we verified E_m to have probability $2^{-0.29}$ (More specifically, $2^{-0.29} \pm 2^{-10.02}$ based on 2^{20} trials and 99% confidence interval). The resulting 36-round boomerang distinguisher holds with probability $2^{(-14.35\times2)+(-15.13\times2)-0.29} = 2^{-59.25}$.

To obtain further improvements, we can consider all E_0 trails that end with (0400,0004,****,****) and all E_1 trails that begin with (8200,****, ****,8000), where * denotes *free* nibbles that can assume any value. The nibbles with fixed differences are the ones involved in the ADD and SUB operations. We denote these as *active* words or nibbles. By considering all valid switches that have this difference propagation, we now have a rectangle distinguisher. This leads to E_0 and E_1 having differential probabilities of $2^{-9.22}$ and $2^{-14.2}$ respectively. The improved 36-round rectangle distinguisher holds with probability $2^{(-9.22\times2)+(-14.2\times2)-0.29} = 2^{-47.13}$.

37-round rectangle distinguisher: Using the same strategy, we first construct a cluster of 18-round E_0 trails (8200,0100, 0001,8000) $\rightarrow$ (0004,0502, ****,****) and a new cluster of 18-round E_1 trails (9000,****, ****,4000) $\rightarrow$ (0100,0281,0002,0000) that begin at an even-numbered round. E_0 and E_1 have differential probabilities of $2^{-11.42}$ and $2^{-14.61}$ respectively. The switching probability of E_m is $ABCT(\texttt{0004}, ROTL_1(\texttt{0502}), ROTR_8(\texttt{4000}), ROTL_1(\texttt{9000}))/2^{32} \approx 2^{31.49-32} = 2^{-0.51}$. Experimentally, this was verified to be $2^{-0.51} \pm 2^{-9.77}$ using 2^{20} trials. The 37-round rectangle distinguisher has a probability of $2^{(-11.42\times2)+(-14.61\times2)-0.51} = 2^{-52.57}$.

38-round rectangle distinguisher: We first construct a cluster of 19-round E_0 trails (8004,4082,8200,0100) $\rightarrow$ (0400 0004,****,****) with a differential probability of $2^{-12.54}$ and reuse the cluster of 18-round E_1 trails from the 36-round rectangle distinguisher. E_m has a switching probability of $2^{-0.28}$, verified using both the ABCT and experimentally. The distinguishing probability of the final 38-round rectangle distinguisher is $2^{(-12.54\times2)+(-14.2\times2)-0.28} = 2^{-53.76}$.

39-round rectangle distinguisher: We reuse the 19-round E_0 cluster from the 38-round distinguisher and search for a 19-round E_1 cluster of trails starting from an odd-numbered round that conforms to (8200,****,****,8000) $\rightarrow$ (0502,0088,0000,000A) . This cluster has a differential probability of $2^{-16.95}$ while the switching probability for E_m is $2^{-0.29}$. The overall 39-round rect-

angle has a distinguishing probability of $2^{(-12.54\times2)+(-16.95\times2)-0.29} = 2^{-59.27}$.

Although it is also possible to construct boomerang distinguishers using a multi-round deterministic boomerang switch, it would impose strong restrictions on the possible output and input differences of E_0 and E_1 respectively. We will not be able to take advantage of the truncated boomerang switch to build larger E_0 and E_1 trail clusters that have higher probabilities. For example, we found that a 4-round deterministic switch would limit β and γ differences to only one active word each, which then leads to shorter E_0 and E_1 trails. This restriction also leads to sub-optimal E_0 and E_1 trails that would further lower the overall rectangle probability. With that said, we provide an example of how a deterministic boomerang switch can still be used to formulate a distinguisher as follows:

An alternative 36-round rectangle distinguisher: Consider a 1-round deterministic switch where β = (****,****,****,****) and γ = (0000,****, ****,0000). For E_0, we build a cluster of 20-round trails that conforms to (8004,4082,8200 ,0100) $\rightarrow$ (****,****,****,****) . The resulting cluster has a differential probability of $2^{-13.62}$. For E_1, we find all 15-round trails (starting from an even-numbered round) that correspond to $\gamma \rightarrow$ (0000,0005,8502,0004) and obtain a cluster of trails with a differential probability of $2^{-16.28}$. The overall 36-round rectangle distinguisher holds with probability $2^{-59.8}$.

8.2. Related-key Boomerang Distinguishers for CHAM

As the designers of CHAM have noted, there is no observable differential effect or trail clustering for related-key differential trails. Therefore, their original 41-round boomerang distinguisher has a probability of 2^{-62}, under the ideal assumption that the boomerang switch is valid and deterministic. We will now show how to adopt the strategy described in Subsection 8.1 to find improved related-key rectangle distinguishers for CHAM. Details for the distinguishers are summarized in Table 9.

41-round rectangle distinguisher: We use a 1-round boomerang switch. E_0 and E_1 of 20 rounds have probabilities of $2^{-11.15}$ and $2^{-13.39}$ respectively. To calculate the probability for E_m using the ABCT, we must consider the 21st round key differences coming from both the upper and lower trail, which in this particular instance are both 0. The probability that the boomerang

41 Rounds	
r_0=20, r_m=1, r_1=20, $p = 2^{-11.15}$, $r = 2^{-0.2}$, $q = 2^{-13.39}$	
ΔK_{E_0}=(0000,0000,0000,0000,0000,4000,4000,0000)	
ΔK_{E_1}=(0080,0000,0000,8000,0000,0000,0000,0000)	
E_0	(8080,4040,4040,0000)→(4200,0000,****,****)
E_m	(4200,0000,****,****)→(8400,****,****,0000)
E_1	(8400,****,****,0000)→(0000,0400,0105,8080)

42 Rounds	
r_0=20, r_m=1, r_1=21, $p = 2^{-11.15}$, $r = 2^{-0.2}$, $q = 2^{-15.39}$	
E_1	(8400,****,****,0000)→(0400,0105,8080,0100)

43 Rounds	
r_0=21, r_m=1, r_1=21, $p = 2^{-10.37}$, $r = 1$, $q = 2^{-15.75}$	
E_0	(8080,4040,4040,0000)→(0000,0000,****,****)
E_m	(0000,0000,****,****)→(****,****,****,****)
E_1	(****,****,****,****)→(0105,8080,0300,0B82)

44 Rounds	
r_0=21, r_m=1, r_1=22, $p = 2^{-10.37}$, $r = 1$, $q = 2^{-19.57}$	
E_1	(****,****,****,****)→(8080,0300,0B82,030B)

Table 9: Related-key rectangle distinguishers for CHAM. The 42-round rectangle uses E_0 and E_m from the 41-round rectangle while the 44-round rectangle uses E_0 and E_m from the 43-round rectangle.

switch occurs is calculated as $ABCT(\texttt{4200}, ROTL_1(\texttt{0000}), ROTR_8(\texttt{8400}), ROTL_1(\texttt{0000})/2^{32} = 2^{-0.2}$. The overall 41-round related-key rectangle distinguisher has a probability of $2^{-49.28}$.

42-round rectangle distinguisher: We append one additional round to E_1 while all other parameters remain. 21 rounds of E_1 holds with probability $2^{-15.39}$. The 42-round related-key rectangle distinguisher has a probability of $2^{-53.28}$.

43-round rectangle distinguisher: The 43-round rectangle distinguisher has the form $(r_0, r_m, r_1) = (21, 1, 21)$. E_0 has a differential probability of $2^{-10.37}$. Since the first two output difference nibbles of E_0 are 0, the ADD operation is not active in E_m, which makes the boomerang switch deterministic regardless of the input difference of E_1. We can then remove all restrictions on the input difference of E_1 and build a cluster of 21-round trails with a differential probability of $2^{-15.75}$. The final 43-round related-key rectangle distinguisher has a probability of $2^{-52.24}$.

44-round rectangle distinguisher: We append one additional round to E_1 while all other parameters remain. 22 rounds of E_1 has a probability of $2^{-19.57}$. The 44-round related-key rectangle has a distinguishing probability of $2^{-59.88}$.

Next, we attempt to construct a rectangle distinguisher using a multi-round switch. Due to the complexity of evaluating multiple rounds of E_m using ABCT when key differences are involved, we will rely on an experimentally derived switching probability. For a relatively few E_m rounds (< 10), the switching probability is high enough to be reliably estimated[3]. First, we search for the best E_0 and E_1 trail then check if the boomerang switch in E_m is valid (or has sufficiently high probability). Then, by fixing the input and output differences of E_0 and E_1, we find all other β and γ combinations where the boomerang switch is valid. This forms a rectangle distinguisher with higher probability than the initial one. Using this strategy, we were able to construct a 46-round related-key rectangle distinguisher:

46-round rectangle distinguisher: We start with a 20-round E_0 related-key trail (8080,4040,4040,0000) $\rightarrow$ (4200,0000, 0000,0084) with a differential probability of 2^{-15}. After 7 rounds of E_m, we append a 19-round E_1

[3]We found that the boomerang switches were either invalid or have switching probabilities larger than 2^{-20} when E_m has fewer than 10 rounds.

related-key trail (0000,0000,0000,8401) $\rightarrow$ (0400,0105,8080,0100) that holds with probability 2^{-12}. E_m was found to have a switching probability of $2^{-9.49}$. Note that for E_0 and E_1, there is no significant differential effect and cannot be individually improved by clustering additional trails. The overall 46-round related-key boomerang distinguisher holds with probability $2^{-63.49}$.

We then find other E_0 and E_1 trails where β and γ describe a valid differential switch. Note that the α = 8080,4040,4040,0000 and δ = 0400,0105,8080,0100 remain unchanged. We sort the additional trails based on their probabilities and consider all possible combinations. After considering more than 1500 combinations (after which improvements to the rectangle probability were negligible), the overall rectangle probability was improved to $2^{-62.78}$.

9. Conclusion

In this paper, we evaluated the security of lightweight block ciphers that employ the ARX and AND-RX constructions against boomerang-style attacks. We first introduced an automated search strategy that leverages the boomerang connectivity table for AND ($\wedge$BCT) to find boomerang and rectangle distinguishers for AND-RX ciphers. We verified the correctness of our automated boomerang and rectangle search on the 32-bit variants of KATAN and SIMON. For the AND-RX cipher KATAN, we found single-key rectangle distinguishers of up to 86, 83 and 62 rounds for the 32, 48 and 64-bit variants respectively. We also found single-key rectangle distinguishers for another AND-RX cipher, SIMON of up to 13 and 16 rounds for the 32 and 48-bit variants respectively. Next, we examined the ARX cipher, CHAM and found that we could formulate a truncated boomerang switch based on the properties of the boomerang connectivity table for ADD (ABCT). This allowed us to find single-key and related-key rectangle distinguishers for CHAM of up to 39 and 46 rounds respectively. All boomerang and rectangle distinguishers are currently the best (longest) distinguishers to date in their respective settings. Our findings are summarized in Tables 1, 2 and 3.

CRediT Authorship Contribution Statement

Li Yu: Conceptualization, Methodology, Validation, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review and

Editing, Visualization; **Je Sen Teh:** Conceptualization, Methodology, Resources, Validation, Investigation, Writing - Original Draft, Writing - Review and Editing, Supervision;

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The full authenticated version of this paper has been published in the Journal of Information Security and Applications (https://doi.org/10.1016/j.jisa.2024.103950). This version corrects two minor typographical errors in the published manuscript – the labelling of rotations in Figure 6 (pg. 28) and the ABCT switch pattern in the 41-round related-key rectangle distinguisher (pg. 31).

References

[1] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, A. Martynenko, IoT, big data, and artificial intelligence in agriculture and food industry, IEEE Internet of Things Journal 9 (9) (2022) 6305–6324. doi:10.1109/jiot.2020.2998584.
URL http://dx.doi.org/10.1109/JIOT.2020.2998584

[2] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, D. Saha, Internet of things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities, IEEE Internet of Things Journal 8 (13) (2021) 10474–10498. doi:10.1109/jiot.2021.3062630.
URL http://dx.doi.org/10.1109/JIOT.2021.3062630

[3] M. Hasan, D. Chang, Lynx: Family of lightweight authenticated encryption schemes based on tweakable blockcipher, IEEE Internet of Things Journal 11 (8) (2024) 14357–14369. doi:10.1109/jiot.2023.3344677.
URL http://dx.doi.org/10.1109/JIOT.2023.3344677

[4] M. Sonmez Turan, K. McKay, D. Chang, L. E. Bassham, J. Kang, N. D. Waller, J. M. Kelsey, D. Hong, Status report on the final round of the NIST lightweight cryptography standardization process, 2023. doi:10.6028/nist.ir.8454.
URL http://dx.doi.org/10.6028/NIST.IR.8454

[5] D. Wagner, The boomerang attack, in: Fast Software Encryption: 6th International Workshop, FSE'99 Rome, Italy, March 24–26, 1999 Proceedings, Springer, 2001, pp. 156–170.

[6] E. Biham, O. Dunkelman, N. Keller, The rectangle attack—rectangling the Serpent, in: Advances in Cryptology—EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings 20, Springer, 2001, pp. 340–357.

[7] O. Dunkelman, N. Keller, A. Shamir, A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony, J. Cryptol. 27 (4) (2014) 824–849.

[8] C. Cid, T. Huang, T. Peyrin, Y. Sasaki, L. Song, Boomerang connectivity table: A new cryptanalysis tool, in: EUROCRYPT (2), Vol. 10821 of Lecture Notes in Computer Science, Springer, 2018, pp. 683–714.

[9] L. Song, X. Qin, L. Hu, Boomerang connectivity table revisited. application to SKINNY and AES, IACR Trans. Symmetric Cryptol. 2019 (1) (2019) 118–141.

[10] H. Wang, T. Peyrin, Boomerang switch in multiple rounds. application to AES variants and deoxys, IACR Trans. Symmetric Cryptol. 2019 (1) (2019) 142–169.

[11] H. Boukerrou, P. Huynh, V. Lallemand, B. Mandal, M. Minier, On the Feistel counterpart of the boomerang connectivity table introduction and analysis of the FBCT, IACR Trans. Symmetric Cryptol. 2020 (1) (2020) 331–362.

[12] H. Hadipour, N. Bagheri, L. Song, Improved rectangle attacks on SKINNY and CRAFT, IACR Trans. Symmetric Cryptol. 2021 (2) (2021) 140–198.

[13] J. S. Teh, A. Biryukov, Differential cryptanalysis of WARP, Journal of Information Security and Applications 70 (2022) 103316. doi:10.1016/j.jisa.2022.103316.
URL http://dx.doi.org/10.1016/j.jisa.2022.103316

[14] V. Lallemand, M. Minier, L. Rouquette, Automatic search of rectangle attacks on Feistel ciphers: Application to WARP, IACR Trans. Symmetric Cryptol. 2022 (2) (2022) 113–140.

[15] H. Hadipour, M. Nageler, M. Eichlseder, Throwing boomerangs into feistel structures: Application to CLEFIA, WARP, LBlock, LBlock-s and TWINE, Cryptology ePrint Archive, Paper 2022/745, https://eprint.iacr.org/2022/745 (2022).
URL https://eprint.iacr.org/2022/745

[16] D. Wang, B. Wang, S. Sun, SAT-aided automatic search of boomerang distinguishers for ARX ciphers, IACR Transactions on Symmetric Cryptology (2023) 152–191.

[17] X. Bonnetain, V. Lallemand, On boomerang attacks on quadratic feistel ciphers new results on KATAN and Simon, IACR Trans. Symmetric Cryptol. 2023 (3) (2023) 101–145.

[18] J. Chen, J. S. Teh, C. Su, A. Samsudin, J. Fang, Improved (related-key) attacks on round-reduced KATAN-32/48/64 based on the extended boomerang framework, in: Information Security and Privacy: 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II 21, Springer, 2016, pp. 333–346.

[19] F. Abed, E. List, S. Lucks, J. Wenzel, Differential cryptanalysis of round-reduced Simon and Speck, in: Fast Software Encryption: 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers 21, Springer, 2015, pp. 525–545.

[20] S. Murphy, The return of the cryptographic boomerang, IEEE Trans. Inf. Theory 57 (4) (2011) 2517–2521. doi:10.1109/TIT.2011.2111091.
URL https://doi.org/10.1109/TIT.2011.2111091

[21] C. De Canniere, O. Dunkelman, M. Knezevic, KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers,

in: Cryptographic Hardware and Embedded Systems-CHES 2009: 11th International Workshop Lausanne, Switzerland, September 6-9, 2009 Proceedings, Springer, 2009, pp. 272–288.

[22] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, The SIMON and SPECK lightweight block ciphers, in: Proceedings of the 52nd annual design automation conference, 2015, pp. 1–6.

[23] B. Koo, D. Roh, H. Kim, Y. Jung, D. Lee, D. Kwon, CHAM: A family of lightweight block ciphers for resource-constrained devices, in: ICISC, Vol. 10779 of Lecture Notes in Computer Science, Springer, 2017, pp. 3–25.

[24] A. Biryukov, D. Khovratovich, Related-key cryptanalysis of the full AES-192 and AES-256, in: Advances in Cryptology–ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings 15, Springer, 2009, pp. 1–18.

[25] R. Ankele, S. Kölbl, Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis, in: SAC, Vol. 11349 of Lecture Notes in Computer Science, Springer, 2018, pp. 163–190.