On Boomerang Attacks on Quadratic Feistel Ciphers

Xavier Bonnetain, Virginie Lallemand

IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 3, pp. 101–145. DOI:10.46586/tosc.v2023.i3.101-145

On Boomerang Attacks on Quadratic Feistel Ciphers

New results on KATAN and Simon

Xavier Bonnetain and Virginie Lallemand

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
xavier.bonnetain@inria.fr,virginie.lallemand@loria.fr

Abstract. The recent introduction of the Boomerang Connectivity Table (BCT) at Eurocrypt 2018 revived interest in boomerang cryptanalysis and in the need to correctly build boomerang distinguishers. Several important advances have been made on this matter, with in particular the study of the extension of the BCT theory to multiple rounds and to different types of ciphers.
In this paper, we pursue these investigations by studying the specific case of quadratic Feistel ciphers, motivated by the need to look at two particularly lightweight ciphers, KATAN and Simon. Our analysis shows that their light round function leads to an extreme case, as a one-round boomerang can only have a probability of 0 or 1. We identify six papers presenting boomerang analyses of KATAN or Simon and all use the naive approach to compute the distinguisher's probability. We are able to prove that several results are theoretically incorrect and we run experiments to check the probability of the others. Many do not have the claimed probability: it fails distinguishing in some cases, but we also identify instances where the experimental probability turns out to be better than the claimed one.
To address this shortfall, we propose an SMT model taking into account the boomerang constraints. We present several experimentally-verified related-key distinguishers obtained with our new technique: on KATAN32 a 151-round boomerang and on Simon-32/64 a 17-round boomerang, a 19-round rotational-xor boomerang and a 15-round rotational-xor-differential boomerang.
Furthermore, we extend our 19-round distinguisher into a 25-round rotational-xor rectangle attack on Simon-32/64. To the best of our knowledge this attack reaches one more round than previously published results.

Keywords: Boomerang attack · Automatic tool · Feistel cipher · KATAN · Simon

1 Introduction

Boomerang cryptanalysis [Wag99] is a variant of differential cryptanalysis [BS91] that was introduced by David Wagner in 1999. It rapidly led to several remarkable results, most notably the full break of AES-192 and 256 in the related-subkeys scenario [BK09].

A boomerang distinguisher corresponds to a couple of differences (α, δ) for which the probability that the following relation is satisfied for a message M is higher for the (reduced) cipher E than for a random permutation:

$$E^{-1}(E(M) \oplus \delta) \oplus E^{-1}(E(M \oplus \alpha) \oplus \delta) = \alpha.$$

The first proposed technique to build such a distinguisher glues together two differential characteristics, as depicted in Figure 1. Following works [BK09, Mur11, Kir15] revealed discrepancies between the theoretical probability predicted by this technique and the

Received: 2023-03-01 Revised: 2023-06-01 Accepted: 2023-08-01 Published: 2023-09-19

actual probability, that could be either in favor or in disfavor of the attacker. A series of works thus started studying how to adjust the distinguisher search method, starting with the sandwich framework [DKS10], and lately with the BCT theory [CHP+18]. Several extensions of this last theory were later proposed, as the counterpart for Feistel ciphers called the FBCT [BHL+20], or the Double Boomerang Connectivity Table [YSS+22].

Table 1: Previous and new boomerang results on the ciphers of the KATAN family. SK stands for the single key scenario, and RK means related-key. Note that the boomerang attack is not the technique leading to the best results on KATAN as 206 rounds of KATAN32 were claimed broken with a single key multidimensional MITM [RR16] for instance.

Cipher	Technique	Att. rds	Distinguisher Probability	Attack Time	Attack Data	Ref.
KATAN32 (254 rds)	Boomerang (SK)	117⊛	-	$2^{79.3}$	$2^{27.3}$	[CTS+16]
	Boomerang (RK)	140†	$2^{-27.2}$	-	-	[ISC13]
		140†	$2^{-26.6}$	-	-	[CTS+16]
		140▷	$2^{-22.04}$	-	-	[JRS22]
		140	2^{-17}	-	-	**Sec. 5.3**
		151	2^{-31}	-	-	**Sec. 5.3**
		154⊛	$2^{-29.72}$	-	-	[CTS+16]
		173†	-	$2^{77.5}$	$2^{27.6}$	[ISC13]
		174†🍃	-	$2^{78.8}$	$2^{27.6}$	[ISC13]
		187⊛ 🍃	-	$2^{78.4}$	$2^{31.8}$	[CTS+16]
KATAN48 (254 rds)	Boomerang (SK)	87	-	2^{78}	$2^{36.7}$	[CTS+16]
	Boomerang (RK)	119†	$2^{-38.8}$	-	-	[ISC13]
		126⊛	$2^{-46.4}$	-	-	[CTS+16]
		145†🍃	-	$2^{78.5}$	$2^{38.4}$	[ISC13]
		150⊛	-	$2^{77.6}$	$2^{47.2}$	[CTS+16]
KATAN64 (254 rds)	Boomerang (SK)	72	-	2^{78}	$2^{55.1}$	[CTS+16]
	Boomerang (RK)	113†	$2^{-52.1}$	-	-	[ISC13]
		116†	$2^{-50.84}$	-	-	[CTS+16]
		130†🍃	-	$2^{78.1}$	$2^{53.1}$	[ISC13]
		133†🍃	-	$2^{78.5}$	$2^{58.4}$	[CTS+16]

In what follows, we prove that the results with a † are invalid. The results with a 🍃 have a time complexity exceeding the generic related-key attack cost. The ones with a ⊛ are invalid according to our experimental verifications and the ones with a ▷ have an experimental probability significantly higher than the theoretical estimate.

Our contributions. We propose a theoretical model for boomerangs of quadratic Feistel ciphers that we generalize to the rotational-xor boomerangs case and the recently introduced rotational-xor-differential boomerangs. These theories are then used to prove many previously published distinguishers on KATAN and Simon are invalid.

We then construct an SMT model to find boomerang distinguishers on KATAN32 and Simon-32/64. We obtained related-key distinguishers for up to 151 rounds of KATAN32 and 17 rounds of Simon-32/64, rotational-xor related-key distinguishers for up to 19 rounds of Simon-32/64 and rotational-xor differential related-key distinguishers for up

Table 2: Summary of previous and new results on the SIMON family. SK: single-key. RK: related-key. CP: chosen plaintexts, FC: full codebook. ID: impossible differential. ZC: zero-correlation. RX: rotational-xor.

SIMON Version	Technique	Att. rds	Distinguisher Probability	Attack Time	Attack Data	Ref.
32/64 (32 rds)	ID (SK)	20	-	$2^{62.8}$	2^{32} (FC)	[DF16]
	ZC (SK)	21	-	$2^{59.42}$	2^{32} (FC)	[SFW15]
	Linear (SK)	23	-	$2^{56.5}$	$2^{31.19}$ CP	[CW16]
	Integral (SK)	24	-	2^{63}	2^{32} (FC)	[CCW+18]
	Differential (SK)	14	$2^{-30.76}$	-	-	[LLW17]
		22	-	$2^{58.76}$	2^{32} (FC)	[QHS15]
	RX (RK)	14	2^{-32}	-	-	[LLA+22]
	RX-differential Rectangle (RK)	15	$2^{-31.32}$	-	-	[CZX+23]
		15	2^{-28}	-	-	**Sec. 5.4.2**
		16	$2^{-31.98}$	-	-	[CZX+23]
	Rectangle (RK)	**12**	1	-	-	**Sec. 5.4.3**
		17†	$2^{-26.72}$	-	-	[ALLW13]
		17	2^{-25}	-	-	**Sec. 5.4.3**
		18†	-	$2^{54.55}$	$2^{30.86}$ CP	[ALLW13]
	RX-Rectangle (RK)	**13**	1	-	-	**Sec. 5.4.1**
		16†	2^{-24}	-	-	[KJK20]
		18	2^{-24}	-	-	**Sec. 5.4.1**
		19	2^{-30}	-	-	**Sec. 5.4.1**
		22†	-	$2^{60.4}$	$2^{30.5}$ CP	[KJK20]
		24	-	$2^{54.6}$	2^{31} CP	**Sec. 6.3**
		25	-	$2^{59.7}$	2^{34} (FC)	**Sec. 6.4**
48/72 (36 rds)	RX-Rectangle (RK)	16▷	2^{-42}	-	-	[KJK20]
		21▷	-	$2^{69.1}$	2^{47} CP	[KJK20]
48/96 (36 rds)	RX-Rectangle (RK)	18▷	2^{-40}	-	-	[KJK20]
		24▷	-	$2^{92.3}$	$2^{46.5}$ CP	[KJK20]
64/96 (42 rds)	RX-Rectangle (RK)	17†	2^{-54}	-	-	[KJK20]
		22†	-	$2^{91.8}$	2^{62} CP	[KJK20]
64/128 (44 rds)	RX-Rectangle (RK)	19†	2^{-52}	-	-	[KJK20]
		25†	-	$2^{123.0}$	$2^{61.5}$ CP	[KJK20]

In what follows, we prove that the results marked by † are invalid while the distinguishers with ▷ have an experimental probability significantly higher than theoretically expected.

to 15 rounds of Simon-32/64. All our distinguishers are experimentally validated. Our results on KATAN are summarized in Table 1 and our results on Simon are summarized in Table 2.

As a side result, we also managed to characterize permutations built from functions using additional linear operations, and obtained that Feistel ciphers are the only possible generic construction, up to affine equivalence. This suggests our theory only applies to ciphers affine-equivalent to a Feistel cipher.

Outline. The necessary preliminaries on boomerang attacks and on KATAN and Simon are presented in the next section, together with an overview of the previous boomerang analyses of these ciphers. In Section 3 we investigate the required criteria for a one-round boomerang to come back when the cipher under study has a Feistel structure with a quadratic round function. We demonstrate that it is a very specific case as the state values play no role in it. We extend this notion to various types of boomerang variants, and we apply our new theory in Section 4 to check previous works. Many of the theoretically predicted probabilities of the naively built distinguishers turn incorrect, which lead us to propose a new model to address this problem in Section 5. New distinguishers of KATAN32 and Simon-32/64 are proposed, and we turn our two best distinguishers on Simon-32/64 into rectangle attacks in Section 6.

2 Preliminaries

2.1 Boomerang Attacks

The boomerang technique was introduced by David Wagner in [Wag99]. It studies quartets of messages and looks for two differences α and δ relating them so that the following event is of higher probability for the cipher E than for a random n-bit permutation:

$$E^{-1}(E(M) \oplus \delta) \oplus E^{-1}(E(M \oplus \alpha) \oplus \delta) = \alpha. \tag{1}$$

In the first place, the technique used to find such a distinguisher was to split the cipher in two ($E = E_1 \circ E_0$) and to find a high probability differential for each part. Assuming we have a differential $\alpha \xrightarrow{E_0} \beta$ with probability p over E_0 and a differential $\delta \xrightarrow{E_1^{-1}} \gamma$ with probability q over E_1^{-1}, the probability of the boomerang distinguisher (1) is expected to be close to p^2q^2. A representation of such a construction is given in Figure 1.

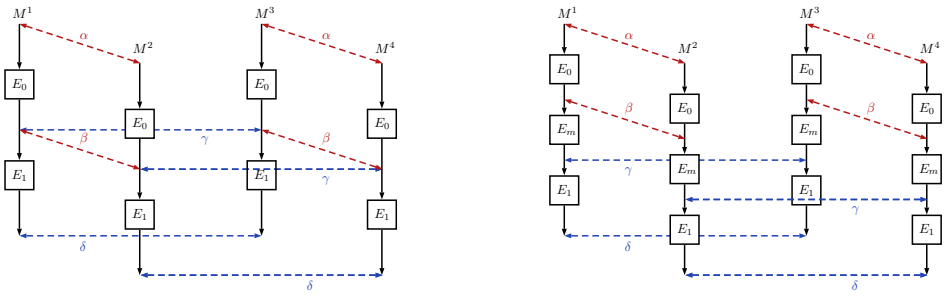

Figure 1: Basic construction of a boomerang distinguisher (left) and sandwich construction (right).

Variants of the boomerang distinguisher. The basic boomerang distinguisher requires chosen plaintexts and adaptative chosen ciphertexts queries: the attacker asks for the encryption of M^1 ($E(M^1) = C^1$) and of $M^2 = M^1 \oplus \alpha$ ($E(M^2) = C^2$), and computes two new ciphertexts from these ($C^3 = C^1 \oplus \delta$, $C^4 = C^2 \oplus \delta$). They next ask for the corresponding plaintexts, and test whether $E^{-1}(C^3) \oplus E^{-1}(C^4) = \alpha$.

Two independent papers introduced variants that get rid of the decryption queries: it was named the amplified boomerang in [KKS01] and the rectangle attack in [BDK01]. In these, an attacker only makes encryption queries (they ask for many pairs satisfying $M^{2i} \oplus M^{2i+1} = \alpha$), and count how many quartets $(M^{2i}, M^{2i+1}, M^{2j}, M^{2j+1})$ meeting the two relations $C^{2i} \oplus C^{2j} = \delta$ and $C^{2i+1} \oplus C^{2j+1} = \delta$ are obtained.

A rather natural extension that was proposed later is the related-key variant [BDK05, HKLP05]. As one would expect, the basic construction of such a distinguisher relies on two related-key differentials, one for E_0 and one for E_1. If we denote by Δ_K^1 the master key difference that is required for the related-key differential over E_0 and by Δ_K^2 the one required for the related-key differential over E_1 the set of 4 keys K^1, K^2, K^3, K^4 that are used to encrypt M^1, M^2, M^3 and M^4 are related by: $K^2 = K^1 \oplus \Delta_K^1$, $K^3 = K^1 \oplus \Delta_K^2$ and $K^4 = K^1 \oplus \Delta_K^1 \oplus \Delta_K^2$.

As we are going to detail in Section 2.5.2 and 2.5.3, [KJK20] and [CZX+23] defined variants of the boomerang distinguisher based on rotational-xor differences instead of standard differences.

Better probability estimates. Several works [BK09, Mur11, Kir15] showed that the actual probability might fall far from the naive estimate of p^2q^2 as the underlying assumption of independence is wrong. Two notable techniques have been introduced to analyze this deviation: the sandwich framework [DKS10] by Dunkelman *et al.* that considers a middle part E_m to isolate the junction between E_0 and E_1 and study their interactions, and the Boomerang Connectivity Table (BCT) by Cid *et al.* [CHP+18]. The BCT is a two-dimensional array that stores the number of solutions of Equation (1) when E is one S-box of the cipher, and can be used to compute the probability of a boomerang over one SPN round. Further extensions of these theories allowed to get more precise estimates of the exact probability of a boomerang, and include these in automatic tools looking for the best distinguishers [DDV20].

2.2 Specification of KATAN

KATAN and KTANTAN [DDK09] are two families of block ciphers that were proposed at CHES 2009 to fit constrained environments, with in particular the goal to be compact in hardware. The two families define three 80-bit key ciphers that differ in the state size that can be either 32, 48 or 64 bits. KTANTAN is even more compact than KATAN by having its key hardcoded in the device. It uses a different key schedule in which flaws were found, leading in particular to a practical break based on the meet-in-the-middle technique [BR11].

No such flaws were found in KATAN, and currently only round-reduced versions have been attacked. The state of the art of boomerang attacks against KATAN is recalled in Table 1. In this paper, we focus in particular on KATAN32.

Round function. An illustration of the functioning of KATAN is given in Figure 2: the n-bit plaintext (with $n = 32, 48$ or 64) is split in two registers, L_1 and L_2, and these are modified by 254 rounds corresponding to either one (for $n = 32$), two (for $n = 48$) or three (for $n = 64$) updates in each round. These updates correspond to a shift to the right of the content of the two registers (bit i moves to position $i + 1$), together with the computation of two new bits, in a feedback shift register way. The new bit of L_1 is computed in a

non-linear way from the bits of L_2 by f_b, and the new bit of L_2 is computed in a non-linear way from the bits of L_1 by f_a:

$$\begin{aligned} L_2[0] &\leftarrow f_a(L_1) &&= L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x_4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a, \\ L_1[0] &\leftarrow f_b(L_2) &&= L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6]) \oplus k_b. \end{aligned}$$

The bits used in f_a and f_b depend on the version of KATAN that is considered, and are recalled in Table 3. IR is a constant bit depending on the round and is referred to as the irregular update rule[1] (so that $L_1[x_5]$ is only used in some rounds) and k_a and k_b are two subkey bits.

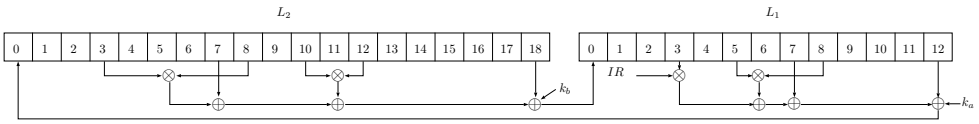

Figure 2: Internal state and functioning of KATAN32.

Table 3: Parameters of the three ciphers of the KATAN family.

Cipher	$\lvert L_1\rvert$	$\lvert L_2\rvert$	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

Key schedule. The round key bits sk are derived from the 80-bit master key k with the help of a LFSR: in round i ($1 \leq i \leq 254$), k_a and k_b correspond to $sk_{2(i-1)}$ and $sk_{2(i-1)+1}$, related by:

$$sk_j = \begin{cases} k_j, & \text{for } 0 \leq j < 80, \\ k_{j-80} \oplus k_{j-61} \oplus k_{j-50} \oplus k_{j-13}, & \text{otherwise.} \end{cases}$$

As an n-bit NLFSR can be seen as a Generalized Feistel Network with n 1-bit wires, the 3 variants of KATAN can be described as quadratic Feistel ciphers, as the Simon ciphers that we now describe.

2.3 Specification of Simon

Simon is a family of lightweight Feistel ciphers proposed together with Speck by Beaulieu *et al.* in [BSS+15]. As depicted in Figure 3, its round function is defined as:

$$f(x) = ((x \lll 8) \cdot (x \lll 1)) \oplus (x \lll 2).$$

There exist 10 variants[2] of Simon, with parameters detailed in Table 4. They differ on the block size ($2n$), on the master key size (made of m words of n bits) and are denoted Simon-$2n/mn$. The key schedule is linear and slightly varies according to the value of m. The case $m = 3$ is depicted on the right in Figure 3, where $(z_j)_i$ represents the ith bit of the

[1] We refer the reader to the specification document for the values of IR in each round.

[2] Five of the larger versions are part of the ISO/IEC standard defining a cryptographic suite for radio frequency identification (RFID) devices (ISO/29167-21:2018).

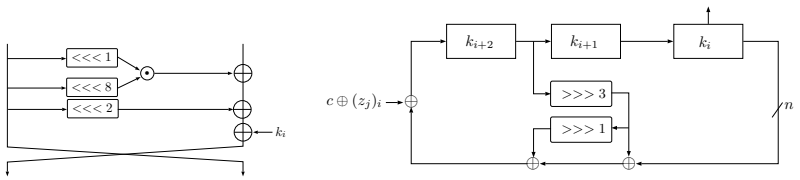

Figure 3: Round function of SIMON (left) and key schedule for the case $m = 3$ key words (right).

constant sequence z_j and where $c = 2^n - 4$. For further details on the specification we refer the reader to [BSS+15]. We present in Table 2 the state of the art of the cryptanalysis[3] of SIMON-32/64 together with the existing boomerang technique results on larger versions. For an overview of the existing analyses on larger versions, one can for instance refer to [LPS21].

Table 4: Parameters of the 10 versions of SIMON.

block size ($2n$)	32	48		64		96		128		
key size (mn)	64	72	96	96	128	96	144	128	192	256
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

2.4 Previous Boomerang Analyses of KATAN

We identified two publications and one ePrint paper that evaluate the resistance of KATAN to boomerang techniques: [ISC13], [CTS+16] and [JRS22], all building distinguishers in the naive way.

2.4.1 Result by Isobe, Sasaki and Chen

In 2013, Isobe *et al.* [ISC13] proposed the first related-key boomerang analysis of KATAN, and described attacks on the 3 variants that can break at least 40 rounds more than the previous techniques (see Table 1). The main idea is to exploit the linearity and simplicity of the key schedule to create *blank* rounds, that are rounds with no differences at all, obtained once the key and state differences cancelled each other. Their main observation is the following:

Observation 1 ([ISC13])**.** *Choosing input key differences properly, 79 consecutive subkey bits have no differences after the key scheduling function.*

In the differential characteristic used for E_0, the difference propagation in the state is chosen so that it cancels out with the key difference after few rounds (these are denoted as *collision step* in [ISC13]). The next rounds form the *blank step* as there is no difference neither in the state nor in the round key. Once the subkey difference introduces a new active bit it corresponds to the *brute force step*, in which the subkey difference propagates to the registers.

[3] As it was never published and we were not able to assess its content, we do not consider the results of [RG18] in Table 2.

A single differential characteristic is considered for the first and second steps, whereas many possible characteristics are considered in the *brute force step*, and taken into account when computing the probability. The differential characteristics used over E_1 are built in a similar manner, with the *collision step* positioned at the ciphertext side, followed by the *blank step* and finally the *brute force step* close to the middle of the cipher. A schematic view of the structure of the boomerang distinguisher used for KATAN32 is given in Figure 4.

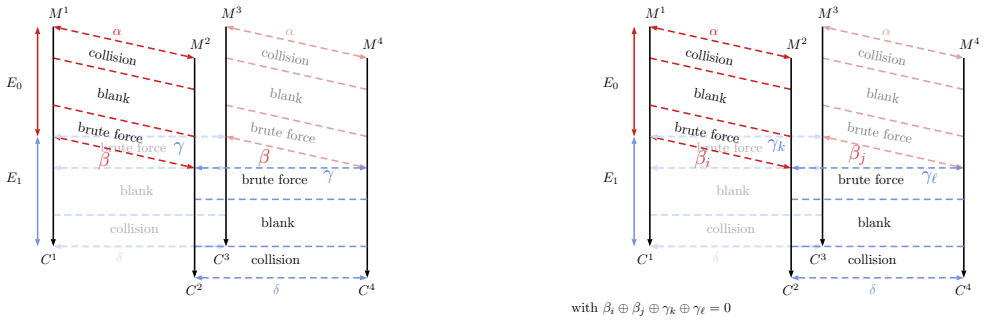

Figure 4: Schematic view of the structure of the boomerang distinguisher used to attack KATAN in [ISC13] (left) and in [CTS+16] (right).

Formally, the probability of the distinguisher is approximated by the formula:

$$\sum_{\beta} Pr^2[\alpha \to \beta] \times \sum_{\gamma} Pr^2[\delta \to \gamma]$$

where β and γ lie in the sets of possible output differences defined by the brute force step.

2.4.2 Result by Chen, Teh, Su, Samsudin and Fang

Three years later and at the same conference, Chen *et al.* [CTS+16] proposed an extension of the work of Isobe *et al.* Their main proposition was to look at what they refer to as an *extended boomerang* distinguisher, that is to consider several differentials for each side of the boomerang (and possibly not the same one for parallel sides), as long as the differences in the middle sum to 0. This construction, already proposed in the seminal work of Wagner [Wag99] is represented on the right of Figure 4 and corresponds to the following probability approximation:

$$\sum_{\beta_i,\beta_j} Pr[\alpha \to \beta_i] \times Pr[\alpha \to \beta_j] \times \sum_{\gamma_k,\gamma_\ell = \beta_i \oplus \beta_j \oplus \gamma_k} Pr[\delta \to \gamma_k] \times Pr[\delta \to \gamma_\ell].$$

To find distinguishers, the authors used a branch-and-bound algorithm to determine the set of differences in which lie the β and γ giving high probability differential characteristics over E_0 and E_1, using the differences found in the previous paper [ISC13] (and in particular the key differences) as a guide. They next built related-key boomerang distinguishers by combining 4 differences summing to 0. Their best result on KATAN32 is a 154-round distinguisher of probability $2^{-29.7209}$ that covers 14 more rounds than the distinguisher given in [ISC13] (140 rounds with probability $2^{-27.2}$). The best attacks claimed in their paper are summarized in Table 1.

2.4.3 Result by Jana, Rahman and Saha

In [JRS22], Jana *et al.* proposed a model to take into account the correlation between AND gates in LFSR-based ciphers. This development led them to a refinement of the

differential bounds of KATAN and TinyJAMBU [WH19] and the technique is also used to propose an improvement over the 140-round distinguisher of Isobe *et al.* The authors look for two 70-round characteristics with their tool and connect them together. After taking into account several high probability characteristics, they conclude in a boomerang distinguisher of probability $2^{-22.04}$.

2.5 Previous Boomerang Analyses of Simon

To the best of our knowledge, there are three articles studying the resistance of Simon to boomerang techniques: [ALLW13], [KJK20] and [CZX+23]. While the first one is a standard boomerang, the latter two use rotational-xor differentials.

2.5.1 Result by Abed, List, Lucks and Wenzel

In an extended version of their FSE paper [ALLW15], Abed *et al.* presented a 17-round related-key boomerang distinguisher of Simon-32/64 [ALLW13, Figure 3]. It relies on two fixed differential characteristics with a two-round cluster in the middle in a similar manner to what was done in [ISC13]. This distinguisher is then extended to an 18-round attack.

2.5.2 Result by Koo, Jung and Kim

The boomerangs from [KJK20] together with the ones presented in [CZX+23] rely on the notion of rotational-xor cryptanalysis, that we now recall.

Rotational cryptanalysis. Rotational cryptanalysis is a technique that is particularly efficient against ARX-based primitives. While similar ideas were used previously, the naming "rotational cryptanalysis" and a formal definition were first proposed in a paper by Khovratovich and Nikolić in 2010 [KN10]. The idea is to study the evolution of a rotational pair $(M, M \lll r)$ through the rounds of the primitive. In 2016, Ashur and Liu [AL16] introduced an extension named the rotational-xor (RX) cryptanalysis, where the pair under study is of the form $(x \oplus a_1, (x \lll \lambda) \oplus a_2)$. This technique was applied to Speck32/64 and is able to distinguish a version reduced to 7 rounds.

Definition 1 (Rotation and rotational-xor difference)**.** As λ is fixed for a given analysis, we denote $\overrightarrow{x} = x \lll \lambda$. Conversely, we denote $\overleftarrow{x} = x \ggg \lambda$. A pair (x, x') has the rotational-xor difference (RXD) α if $x' = \overrightarrow{x} \oplus \alpha$.

Koo *et al.* [KJK20] and Chen *et al.* [CZX+23] applied variations of the rotational-xor cryptanalysis to the non-ARX cipher Simon [BSS+15]. They used $\lambda = 1$, which was shown to be optimal for Simon in [LLA+22].

Definition of the rotational-xor rectangle. One of the contributions of the article of Koo *et al.* [KJK20] is the definition of the rotational-xor rectangle characteristic as the analogue of the usual differential-based rectangle characteristic. The authors propose to construct rotational-xor rectangle distinguishers in the same way as naive rectangle distinguishers: the cipher is split in two ($E = E_1 \circ E_0$) and an RX characteristic is searched for each part: if the RX characteristic $(\alpha \to \beta)$ is satisfied with probability p for E_0 and the RX characteristic $(\gamma \to \delta)$ is satisfied with probability q for E_1 then a rotational-xor rectangle of expected probability $p^2q^2 2^{-n}$ is obtained.

Their theory relies on the following theorem that justifies that two RX characteristics can be connected together to create a rotational-xor rectangle characteristic:

Theorem 1 ([KJK20, Theorem 4])**.** *Let x and y be independent random variables and α, β be constants in $\mathbb{F}_2^n$. Then $(x, \overrightarrow{x} \oplus \alpha)$ and $(y, \overrightarrow{y} \oplus \beta)$ are RX pairs. If $(x, \overrightarrow{y} \oplus \beta)$*

forms an RX pair with RXD γ, then $(y, \overrightarrow{x} \oplus \alpha)$ also forms an RX pair and its RXD is $\delta = \alpha \oplus \beta \oplus \gamma$.

Put differently, this theorem implies that if two RX pairs of messages (M^1, M^2) and (M^3, M^4) follow the RXD characteristic over E_0 (probability p^2) and that $(E_0(M^1), E_0(M^4))$ is an RX pair of RXD γ (probability 2^{-n}) then the pair $(E_0(M^3), E_0(M^2))$ is automatically an RX pair of RXD γ. Following the RXD characteristic over E_1 for these two is then of probability q^2, and assuming independencies between all these events leads to the claimed probability of $p^2q^2 2^{-n}$ (see Figure 5).

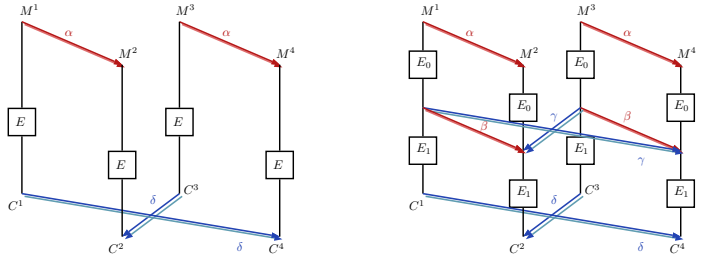

Figure 5: General structure of a rotational-xor boomerang distinguisher (left) and construction from RX differentials proposed in [KJK20] (right). The direction of the arrows gives the RX pairs.

2.5.3 Result by Chen, Zhu, Xiang, Xu, Zeng and Zhang

In an article published at CT-RSA 2023 [CZX+23], Chen *et al.* introduced a new variant of the boomerang distinguisher based on two different types of differential properties, rotational-xor differentials for the top characteristic in E_0, and standard differentials for the bottom characteristic in E_1, with no switch in the middle. An illustration of this type of distinguisher is given in Figure 6, and an overview of the results presented in this article on Simon-32/64 is given in Table 2.

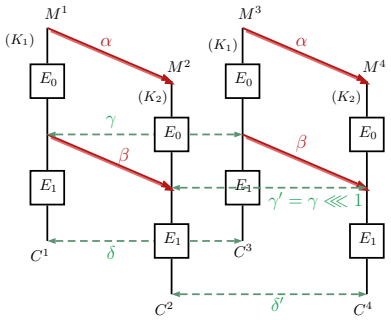

Figure 6: Rotational-xor differential rectangle distinguisher as introduced in [CZX+23]: (M^1, M^2) and (M^3, M^4) are expected to follow a rotational-xor differential characteristic from α to β over E_0, while $(E_0(M^1), E_0(M^3))$ should follow the differential characteristic from γ to δ, and $(E_0(M^2), E_0(M^4))$ should follow the differential characteristic from $\gamma' = \gamma \lll 1$ to δ'.

This technique is particularly relevant in the case of ciphers with non-linear key schedules (the authors of [CZX+23] showed how to apply it to Simeck) as the top part

can be built in the related-key setting, while the bottom part is in the single-key setting. This way, only the RX characteristic implies conditions on the keys, and thus the weak key space is larger and the final attack could be more powerful than a rotational-xor rectangle one. This setting is illustrated in Figure 6, where the key is the same for messages with indexes of the same parity. Doing this, the keys are related in the top characteristics, but are equal for the bottom ones.

Note also that this specific setting implies that the starting point for the bottom differential characteristics is not the same for the two sides. Because of the RX characteristic outputs, we have the following proposition, taken from [CZX+23], and illustrated for $\lambda = 1$ on Figure 6:

Proposition 1 ([CZX+23]). *If $(E_0(M^1), E_0(M^2))$ and $(E_0(M^3), E_0(M^4))$ are RX pairs with the rotation offset λ and the RX-difference β, while the keys satisfy the corresponding key RX-difference, then we have $\gamma' = \gamma \lll \lambda$.*

In the cases studied in [CZX+23], the two differential characteristics are a rotational shift of each other, taking advantage of the fact that the round functions of Simon and Simeck are invariant by rotation.

3 One-Round Boomerangs for a Quadratic Feistel Cipher

Sandwich theory and boomerang tables. As briefly stated in the preliminary section, boomerang distinguishers naively built by combining two differential characteristics frequently have a probability that deviates from the expected one. Cases where the probability is better than estimated arose for instance in [BK09], while incompatibilities were identified by Murphy and Kircanski in [Mur11] and [Kir15]. Following the sandwich framework by Dunkelman *et al.* [DKS10], a better probability approximation is obtained by dividing the distinguisher into three parts: (E_0, E_m, E_1), where E_m encloses the rounds where the top and bottom differential characteristic of E_0 and E_1 intermingle, while E_0 and E_1 are considered as parts where no interdependency happens. The BCT theory introduced in 2018 formalizes the computation of the probability of a middle part E_m made of 1 SPN round, while following researches show how to deal with more rounds, for both SPN [WP19, SQH19] and S-box based Feistel ciphers [BHL+20].

In the case of KATAN and Simon we cannot directly apply the BCT or FBCT theories as we are dealing with bit-oriented ciphers. Thus, we go back to the equations expressing that a quartet comes back over one cipher round (deduced from Equation (1)).

Note that most analyses to date consider boomerang distinguishers with parallel sides, in which the same differential characteristic is followed between M^1 and M^2 and between M^3 and M^4 for the top part and similarly between M^1 and M^3 and between M^2 and M^4 for the bottom part. Unless stated differently, we also focus on this setting.

3.1 Expression of the Conditions for a Quadratic Feistel Cipher

We consider here the generic case of a Feistel round $R(x_\ell, x_r) = (x_r \oplus f(x_\ell), x_\ell)$, and compute the boomerang equation for the input/output differences $((\alpha_\ell, \alpha_r), (\delta_\ell, \delta_r))$. We need that

$$R^{-1}\left(R(x_\ell, x_r) \oplus (\delta_\ell, \delta_r)\right) \oplus R^{-1}\left(R\left((x_\ell, x_r) \oplus (\alpha_\ell, \alpha_r)\right) \oplus (\delta_\ell, \delta_r)\right) = (\alpha_\ell, \alpha_r).$$

This reduces to

$$\begin{aligned}(x_\ell \oplus \delta_r \oplus \alpha_\ell, x_r \oplus f(x_\ell) \oplus \delta_\ell \oplus f(x_\ell \oplus \delta_r) \oplus \alpha_r) = \\ (x_\ell \oplus \delta_r \oplus \alpha_\ell, x_r \oplus \alpha_r \oplus f(x_\ell \oplus \alpha_\ell) \oplus f(x_\ell \oplus \delta_r \oplus \alpha_\ell) \oplus \delta_\ell)\,.\end{aligned}$$

Hence, we have:

Proposition 2 (Boomerang constraint for Feistel ciphers [BHL+20])**.** *A boomerang returns from 1 round of Feistel cipher with round function f for the input x_ℓ, x_r if and only if*

$$f(x_\ell) \oplus f(x_\ell \oplus \delta_r) \oplus f(x_\ell \oplus \alpha_\ell) \oplus f(x_\ell \oplus \delta_r \oplus \alpha_\ell) = 0,$$

that is, the second derivative of f at points α_ℓ, δ_r must be zero.

In Proposition 2, "Feistel" has to be understood in a loose sense, that also encompasses NLFSR and Lai-Massey schemes. We discuss the classes of ciphers for which this proposition is relevant in Appendix A.

Of course, if f is quadratic, the second derivative is constant. That is, depending on α_ℓ and δ_r, the transition probability is either 0 or 1. This peculiar behaviour was evoked in [BHL+20] to compute the Feistel boomerang uniformity of quadratic invertible S-boxes, but was not investigated further.

A quadratic function only contains monomials of degree up to 2. To compute the boomerang constraint of such a function, we start by studying the case of the 1-bit AND, from which more intricate cases can be deduced.

Proposition 3 (Boomerang constraint for a 1-bit AND)**.** *Let $f(x, y) = xy$. The second derivative of f in points (α_x, α_y) and (δ_x, δ_y) is equal to $\alpha_x\delta_y \oplus \alpha_y\delta_x$.*

Proof. The second derivative is equal to

$$xy \oplus (x \oplus \alpha_x)(y \oplus \alpha_y) \oplus (x \oplus \delta_x)(y \oplus \delta_y) \oplus (x \oplus \alpha_x \oplus \delta_x)(y \oplus \alpha_y \oplus \delta_y) = \alpha_x\delta_y \oplus \alpha_y\delta_x.$$

□

From this proposition, it suffices to decompose any quadratic f in a sum of quadratic monomials and to take into account the linear layers to obtain the round constraint on the boomerang characteristic. Below, we apply this to KATAN and Simon.

3.1.1 Application to KATAN

Details for KATAN32. There are two independent non-linear functions in KATAN32:

- For the update of the register L_2: $L_1[5] \cdot L_1[8]$,
- For the update of the register L_1: $L_2[10] \cdot L_2[12] \oplus L_2[3] \cdot L_2[8]$.

We denote by α_i (resp. δ_i) the input (resp. output) difference on bit i of L_2, and α'_i, δ'_i the corresponding difference on bit i of L_1.

For the first function we can directly apply Proposition 3. As the round function shifts bits, bits 5 and 8 of L_1 become, after one round, bits 6 and 9. Hence, we have the following formula:

$$\alpha'_8 \cdot \delta'_6 \oplus \alpha'_5 \cdot \delta'_9 = 0.$$

For the second function, we can proceed the same way. As we have a xor of two AND we need to take the xor of the two corresponding equations. We obtain the following formula:

$$\alpha_{12} \cdot \delta_{11} \oplus \alpha_{10} \cdot \delta_{13} \oplus \alpha_8 \cdot \delta_4 \oplus \alpha_3 \cdot \delta_9 = 0.$$

KATAN32, KATAN48 and KATAN64. By following a similar process for KATAN48 and KATAN64, we obtain that the boomerang comes back over one round if and only if the following equations are satisfied:

For KATAN32:

$$\begin{cases} \alpha_{12} \cdot \delta_{11} \oplus \alpha_{10} \cdot \delta_{13} \oplus \alpha_8 \cdot \delta_4 \oplus \alpha_3 \cdot \delta_9 = 0 \\ \alpha'_8 \cdot \delta'_6 \oplus \alpha'_5 \cdot \delta'_9 = 0 \end{cases}$$

For KATAN48:

$$\begin{cases} \alpha_{20} \cdot \delta_{14} \oplus \alpha_{12} \cdot \delta_{22} \oplus \alpha_{14} \cdot \delta_7 \oplus \alpha_5 \cdot \delta_{16} = 0 \\ \alpha_{21} \cdot \delta_{15} \oplus \alpha_{13} \cdot \delta_{23} \oplus \alpha_{15} \cdot \delta_8 \oplus \alpha_6 \cdot \delta_{17} = 0 \\ \alpha'_{14} \cdot \delta'_8 \oplus \alpha'_6 \cdot \delta'_{16} = 0 \\ \alpha'_{15} \cdot \delta'_9 \oplus \alpha'_7 \cdot \delta'_{17} = 0 \end{cases}$$

And for KATAN64:

$$\begin{cases} \alpha_{12} \cdot \delta_{10} \oplus \alpha_{19} \cdot \delta_{34} \oplus \alpha_{31} \cdot \delta_{22} \oplus \alpha_7 \cdot \delta_{15} = 0 \\ \alpha_{13} \cdot \delta_{11} \oplus \alpha_{20} \cdot \delta_{35} \oplus \alpha_{32} \cdot \delta_{23} \oplus \alpha_8 \cdot \delta_{16} = 0 \\ \alpha_{14} \cdot \delta_{12} \oplus \alpha_{21} \cdot \delta_{36} \oplus \alpha_{33} \cdot \delta_{24} \oplus \alpha_9 \cdot \delta_{17} = 0 \\ \alpha'_{18} \cdot \delta'_{12} \oplus \alpha'_9 \cdot \delta'_{21} = 0 \\ \alpha'_{19} \cdot \delta'_{13} \oplus \alpha'_{10} \cdot \delta'_{22} = 0 \\ \alpha'_{20} \cdot \delta'_{14} \oplus \alpha'_{11} \cdot \delta'_{23} = 0 \end{cases}$$

3.1.2 Application to Simon

As each output bit of the round function of Simon contains only 1 AND, we can directly apply Proposition 3. Moreover, we can gather these equations into one single vectorial constraint:

$$(\alpha_\ell \lll 8) \cdot (\delta_r \lll 1) \oplus (\alpha_\ell \lll 1) \cdot (\delta_r \lll 8) = 0 \tag{2}$$

3.2 RX-Boomerang Variant of the Constraint

For completeness, we provide below the formula of the relation that has to be satisfied for an RX-boomerang to come back for a cipher E. When expressed for one round (or at the S-box level), this formula corresponds to the counterpart of the BCT for the RX scenario.

Theorem 2 (Expression of the rotational-xor boomerang constraint.)**.** *Let (M^1, M^2) be a plaintext RX pair of RXD α and let M^2 and M^3 be two plaintexts built so that $(E(M^1), E(M^4))$ and $(E(M^3), E(M^2))$ are RX pairs of RXD δ. The pair (M^3, M^4) forms an RX pair of RXD α if and only if the following equality holds:*

$$E^{-1}(\overrightarrow{E(M^1)} \oplus \delta) \oplus \overleftarrow{E^{-1}(\overrightarrow{E(M^1} \oplus \alpha) \oplus \delta)} = \alpha$$

The proof is detailed in Appendix B. As with standard boomerangs, we can specialize this property for Feistel ciphers (with notation from Figure 7, left):

Proposition 4 (RX-boomerang constraint for Feistel ciphers)**.** *An RX-boomerang returns from one round of Feistel cipher with round function f for the input x_ℓ, x_r if and only if*

$$\overrightarrow{f(x_\ell)} \oplus f(\overrightarrow{x_\ell} \oplus \delta_r) \oplus f(\overrightarrow{x_\ell} \oplus \alpha_\ell) \oplus \overleftarrow{f(x_\ell \oplus \overleftarrow{\delta_r} \oplus \overleftarrow{\alpha_\ell})} = 0.$$

Remark 1 (Rotation-invariant functions). If f is invariant by rotation, that is, $\overrightarrow{f(x)} = f(\overrightarrow{x})$, then the Feistel RX-boomerang constraint becomes

$$f(\overrightarrow{x_\ell}) \oplus f(\overrightarrow{x_\ell} \oplus \delta_r) \oplus f(\overrightarrow{x_\ell} \oplus \alpha_\ell) \oplus f(\overrightarrow{x_\ell} \oplus \delta_r \oplus \alpha_\ell) = 0,$$

which is the Feistel boomerang constraint up to a rotation of x_ℓ.

This remark is especially helpful for quadratic f: as the input value does not intervene in the constraint, the two cases are strictly equivalent.

Case of Simon. As the inner function of Simon is rotation-invariant, the constraint is the same as in normal boomerangs:

$$(\alpha_\ell \lll 8) \cdot (\delta_r \lll 1) \oplus (\alpha_\ell \lll 1) \cdot (\delta_r \lll 8) = 0 \tag{3}$$

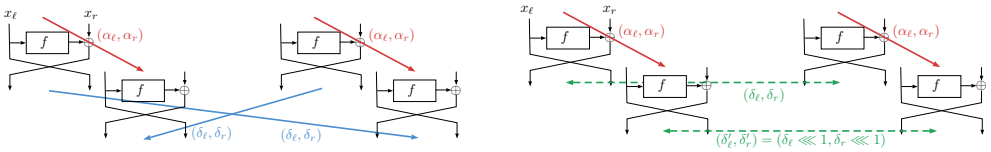

Figure 7: RX-boomerang property (left) and RX-differential boomerang property (right) for a Feistel cipher.

3.3 RX-differential Boomerang Variant of the Constraint

The case of RX-differential boomerangs is analogous to RX-boomerangs, with fewer rotations as the bottom part considers simple differences (see Figure 7, right). We obtain the following proposition.

Proposition 5 (RX-differential boomerang constraint for Feistel ciphers)**.** *An RX-differential boomerang returns from 1 round of Feistel cipher with round function f for the input x_ℓ, x_r if and only if*

$$\overrightarrow{f(x_\ell)} \oplus \overrightarrow{f(x_\ell \oplus \delta_r)} \oplus f(\overrightarrow{x_\ell} \oplus \alpha_\ell) \oplus f(\overrightarrow{x_\ell} \oplus \overrightarrow{\delta_r} \oplus \alpha_\ell) = 0.$$

Remark 2. If f is invariant by rotation, the formula reduces to

$$f(\overrightarrow{x_\ell}) \oplus f(\overrightarrow{x_\ell} \oplus \overrightarrow{\delta_r}) \oplus f(\overrightarrow{x_\ell} \oplus \alpha_\ell) \oplus f(\overrightarrow{x_\ell} \oplus \overrightarrow{\delta_r} \oplus \alpha_\ell) = 0,$$

which is the same as the RX-boomerang constraint, up to a shift of the lower difference.

Case of Simon. Assuming $\overrightarrow{x} = x \lll 1$, the equation is

$$(\alpha_\ell \lll 8) \cdot (\delta_r \lll 2) \oplus (\alpha_\ell \lll 1) \cdot (\delta_r \lll 9) = 0. \tag{4}$$

4 Incompatibilities in the Previous Attacks

On invalid boomerang distinguishers. In this section, we show multiple previous boomerang distinguishers have a flawed analysis, that is, the characteristics presented for the top and bottom parts are either incompatible, or fit better together than what was expected. When reasonning on characteristics, an incompatibility only suggests the probability is likely to be lower than expected, and if all considered characteristics are incompatible, we can expect the distinguisher to have a probability around 2^{-n}. To be able to conclude on the general distinguisher, we also experimentally tested the distinguishers on 32- and 48-bit ciphers.

4.1 Results on KATAN

4.1.1 Attacks on KATAN from [ISC13]

We start by computing the boomerang constraints on one of the fully specified boomerang characteristics. When looking at the middle round defined by the differential characteristics over E_0 and E_1 in [ISC13, Table 11 and 12], we have

$$A = (0,0,1,0,\ 0,0,0,1,\ 1,0,0,0,\ 0,0,0,0,\ 0,0,0;1,\ 0,1,0,0,\ 0,0,0,0,\ 0,0,0,0) \text{ and}$$

$$D = (0,0,0,0,\ 1,0,0,0,\ 0,0,0,0,\ 0,0,0,1,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,1,0,\ 0,0,0,1).$$

Which for the first expression we are interested in gives $\alpha_{12}\delta_{11} \oplus \alpha_{10}\delta_{13} \oplus \alpha_8\delta_4 \oplus \alpha_3\delta_9 = 0 \times 0 \oplus 0 \times 0 \oplus 1 \times 1 \oplus 0 \times 0 = 1$ so the boomerang does not come back over the middle round.

Still, the boomerang attack proposed by Isobe *et al.* relies on a series of characteristics (see Figure 4) so we cannot conclude at this stage that their distinguisher is invalid. However, the high probability differentials rely on blank rounds that must appear, so we know that the state difference entering round 54 in E_0 is equal to

$$\varepsilon = (0,0,0,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,0,1,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,0,0)$$

while the difference entering round 84 in E_1 is

$$\tau = (0,0,0,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,1,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,0,0,0).$$

Given the slow diffusion of KATAN, the knowledge of these fixed differences implies the knowledge of many other bit differences in previous or following rounds. The bit differences that are known with probability 1 in E_0 and E_1 knowing that ε and τ appear are given in Appendix, in Table 15 and Table 16 respectively (unknown bits are denoted "?").

For a not-extended[4] boomerang to be valid, the boomerang constraints that we detailed in Section 3.1.1 have to be fulfilled in every round. By checking these conditions in rounds where enough bits are known to determine the value of the products at play, we are able to conclude that the boomerang is invalid: indeed, in round 69 to 70 we have:

$$\begin{cases} \alpha_{12}\delta_{11} \oplus \alpha_{10}\delta_{13} \oplus \alpha_8\delta_4 \oplus \alpha_3\delta_9 = 0 \times 0 \oplus 0 \times 0 \oplus 1 \times 1 \oplus 0 \times 0 = 1 \neq 0 \\ \alpha'_8\delta'_6 \oplus \alpha'_5\delta'_9 = 0 \times 0 \ \oplus \ ? \times 0 = 0 \end{cases}$$

This implies that no matter which characteristic is considered in the attack, the boomerang constraints cannot be fulfilled.

This observation is quite surprising as the authors of [ISC13] claimed that they practically verified the validity of their 140-round distinguisher and even provided the detail of a right quartet. To understand this contradiction, we studied the quartet provided in their article. By computing the intermediate differences, we observed that the top characteristic (over E_0) is followed, but not the bottom one. We next reproduced their experiment to obtain the experimental probability that the input difference comes back given a boomerang configuration. In our tests, we build pairs with a fixed input difference, compute their ciphertexts after 140 rounds and infer two new ciphertexts from them by adding the ciphertext difference, and finally count how many times their decryptions give the expected input difference. We found out that it happens on average once every 2^{32} tests, which confirms the distinguisher is invalid.

A summary of the contradictions we found is given in Table 5.

Table 5: Boomerang distinguishers given in [ISC13].

Variant	Dist. size (E_0, E_1)	Claimed proba. $(pq)^2$	Round contradiction
32	$70 + 70 = 140$	$(2^{-7.1}2^{-6.5})^2 = 2^{-27.2}$	69-70
48	$60 + 59 = 119$	$(2^{-10.9}2^{-8.5})^2 = 2^{-38.8}$	65-66, 66-67
64	$56 + 57 = 113$	$(2^{-12.25}2^{-13.8})^2 = 2^{-52.1}$	55-56, 61-62, 62-63

[4]That is, following the same differential characteristic over facing sides.

4.1.2 Attacks on KATAN from [CTS+16]

In the same way, we analyze the distinguishers given in [CTS+16] by computing the probability-one propagation of the differences in E_0 and E_1 and checking if the boomerang constraints are fulfilled. In their article the authors consider *extended boomerang distinguishers* (that is, with distinct characteristics on each side) whereas our theoretical analysis only covers the non-extended component of their cluster.

To be able to conclude, we checked experimentally the boomerang probability with a program written in C language. As shown in Table 6, we were able to find theoretical or experimental inconsistencies in all the related-key boomerang distinguishers given in [CTS+16]. The single-key distinguisher on KATAN32 was not confirmed experimentally.

Table 6: Boomerang distinguishers given in [CTS+16], in the related-key setting.

Variant	Dist. size	Claimed proba.	Round contradiction	Experimental proba.
32	$70+70=140$	$2^{-26.58}$	63-64,69-70	$< 2^{-32}$
	$70+71=141$	$2^{-28.58}$	64-65	$< 2^{-32}$
	$70+72=142$	$2^{-30.58}$	78-79	$< 2^{-32}$
	$70+84=154$	$2^{-29.72}$	none found	$< 2^{-32}$
48	$63+63=126$	$2^{-46.4}$	none found	$< 2^{-48\dagger}$
64	$56+60=116$	$2^{-50.84}$	52-53,54-55	

† Only plaintexts of a certain shape can satisfy the characteristic. With 6 fixed bits, the probability is improved by a factor of 2^{12}, and we tested 2^{38} such plaintexts.

4.1.3 Distinguisher on KATAN from [JRS22]

By combining two 70-round differential characteristics found with their model, Jana *et al.* built a naive boomerang distinguisher of theoretical probability equal to $2^{-22.04}$. We experimentally checked this distinguisher and found out that its actual probability is much higher, close to $2^{-15.8}$. While the characteristics were not provided in the paper, we expect this gap to be caused by the inner boomerang constraints.

4.2 Results on Simon

4.2.1 Attack on Simon from [ALLW13]

Many types of distinguishers are presented in [ALLW13], and among them is a 17-round related-key boomerang for Simon-32/64. The boomerang consists in two fully-specified characteristics, each completed by a small two-round cluster in the middle rounds. Hence, we have access to two rounds of connected truncated differentials, and we can check if they are compatible. We found that the transition from round 8 to 9 is impossible for all characteristics in the cluster, as shown in Table 7. We also confirmed experimentally the distinguisher has probability around 2^{-32} instead of the claimed $2^{-26.72}$.

4.2.2 Attacks on Simon from [KJK20]

We study the characteristics given in [KJK20] to see if we spot contradictions. As for KATAN, the authors took advantage of the key (rotational) difference to generate blank

Table 7: Inner differential cluster of the 17-round boomerang taken from [ALLW13, Figure 3], and the corresponding boomerang constraint.

Round	Top left differential	Bottom right differential	Boomerang constraint
8-9	0011000?10000?0	?00010????100???	?0?1000??0000000
9-10	?00??1?0????001?	0000011?10000?0	?000000??00000??

rounds, which are of probability 1. Since these blank rounds are central to get a high probability, we conduct a verification that starts from the round before the blank rounds of the top characteristic (that is, the round that set up the upper blank rounds) to the one after the blank rounds of the lower characteristic (the round that set up the lower blank rounds). We set the upper and lower differences (plus the key differences), and we propagate them with probability 1 over all the rounds. We next apply the boomerang constraint formula (Equation (3)) in each round to see if we find a contradiction.

As detailed in Table 8, we found contradictions in 3 out of the 5 distinguishers given in [KJK20] (as an example, the details for the case of Simon-32/64 are provided in Table 17). The distinguishers on Simon-48/72 and Simon-48/96 (recalled in Table 18) show no contradictions.

Table 8: Rotational-xor boomerang distinguishers given in [KJK20].

Variant	Dist. size (E_0, E_1)	Claimed proba. $(pq)^2$	Experimental proba.	Round contradiction
32/64	$8+8=16$	$(2^{-6}2^{-6})^2 = 2^{-24}$	$< 2^{-32}$	16-17
48/72	$7+9=16$	$(2^{-4}2^{-17})^2 = 2^{-42}$	2^{-36}	none
48/96	$9+9=18$	$(2^{-10}2^{-10})^2 = 2^{-40}$	2^{-27}	none
64/96	$8+9=17$	$(2^{-10}2^{-17})^2 = 2^{-54}$		26-27
64/128	$9+10=19$	$(2^{-10}2^{-16})^2 = 2^{-52}$		16-17, 17-18

Simon-48. As the characteristic is fully specified and there is no cluster in these distinguishers, we can provide manually a better estimate of the probabilities. If the transition of the characteristics over 1 round are with probability p and q and the boomerang constraint passes with probability 1, the 1-round boomerang has probability pq instead of p^2q^2. Here, we can compute the constraint over 2 rounds, and for each round only one differential transition is specified.

The characteristics are recalled in Table 18. For Simon-48/72, the constraints are in rounds 13-14 and 14-15. For both rounds the specified transition has probability 2^{-2}, which means the probability can be expected to be 2^4 times higher than naively computed. For Simon-48/96, the probabilities are 2^{-2} and 2^{-4}, meaning the probability can be expected to be 2^6 times higher than naively computed.

This is of course not sufficient to get an accurate probability: the characteristics can be extended to the rounds above and below, and these rounds could show a contradiction or have a higher probability. Hence, we experimentally tested the probability of these two distinguishers.

In the case of Simon-48/96, we build 2^{36} quartets for each of the 64 picked random keys and we observed a boomerang probability equal to 2^{-27} (instead of the expected 2^{-40}). The gap in the probability can already be observed in the middle rounds, when

Table 9: Some 48-bit RX-differential characteristics from [CZX+23].

Cipher	rds	Input RX-diff	Output diff	α_ℓ	δ_r	Boomerang Constraint	[CZX+23] Tables
Simon48/72	15	(0,3e)	(222,80)	284900	22	8	19, 22
Simon48/72	16	(0,3e)	(222,80)	284900	80	0	19, 22
Simon48/96	16	(0,180016)	(222,80)	800019	22	80	15, 22
Simon48/96	17	(0,180016)	(222,80)	800019	80	0	15, 22
Simeck48	20	(0,110)	(28,10)	6f	28	40	20, 21
Simeck48	21	(0,110)	(28,10)	6f	40	80	20, 21

testing a version of the distinguisher reduced to round 14 to 30 (so capturing the blank rounds and one more round) that experimentally has probability 2^{-15} instead of 2^{-28}.

In the case of Simon-48/72, we checked the differential with 2 less rounds at the end, which is expected to improve the success probability of the boomerang by a factor 2^4. We built 2^{36} quartets for each of the 64 picked random keys and we observed a boomerang probability equal to 2^{-32} (instead of the expected 2^{-38}). Thus, we extrapolate that the actual probability of the full distinguisher is 2^{-36} instead of 2^{-42}.

Overall, all the proposed distinguishers are either invalid, or have a probability much better than expected.

4.2.3 Distinguishers on Simon from [CZX+23]

The article [CZX+23] proposes some distinguishers on all versions of Simon and Simeck with blocks of 32 and 48 bits. The authors experimentally checked the 32-bit distinguishers. However, the 48-bit distinguishers were not experimentally validated. We check some high-probability characteristics used in the distinguishers presented in their Tables 8 and 10 using Equation (4) when the corresponding characteristic was provided in appendix. As the provided top and bottom characteristics do not overlap, we can only check the constraint on 2 rounds. Moreover, the second-to-last-round RX-differential is always 0, meaning we could only check 1 meaningful equation. As shown in Table 9, 4 of the 6 48-bit characteristics we could check were not instantiable.

Note that this only proves the actual pair of characteristics is not instantiable, not the full distinguisher. Moreover, the differential characteristic can be rotated to align the bits correctly and pass the boomerang constraint. On the other hand, a contradiction could arise in another round, and the approach in [CZX+23] relies on clustering, making it more likely that some characteristics involved in the cluster are incompatible. In particular, in the case of 21-round Simeck48, the incompatibility arises between two optimal characteristics in the cluster.

Moreover, while we could not prove it is incorrect, our results in Section 5.4.2 suggests the 16-round Simon-32/64 distinguisher with probability $2^{-31.98}$ proposed in [CZX+23] should be taken with caution.

5 Boomerang Distinguisher Models on KATAN and Simon

5.1 State of the Art

As of the time of writing, two major techniques have been proposed to automatically search for boomerang distinguishers:

The technique by Delaune *et al.* [DDV20]. This technique was first applied on Skinny and later used on the Feistel cipher Warp in [LMR22]. This complex model includes all the possible tables that might define the boomerang probability and automatically decides where to use each table to maximize the final probability. It first searches a truncated solution, while in a second phase a concrete binary definition of the boomerang is searched.

The technique by Hadipour *et al.* [HBS21]. This technique was first proposed to attack Skinny and Craft and was later on slightly modified and applied to Feistel ciphers in [HNE22]. The MILP model devised in these articles takes as input the size of E_0, E_m and E_1 and has as objective function to minimize a weighted sum of the number of active S-boxes in E_0 and E_1 and of the common active S-boxes in E_m. The idea behind this minimization is to choose good differential characteristics for E_0 and E_1 while maximizing the number of ladder switches (that is, cases where the BCT coefficient is the highest) in the middle part E_m. In the version for Feistel ciphers, the propagation of the differences in the middle part E_m are made with probability 1. Again, the first step is made at the truncated level, and a binary solution is searched afterwards. The probability of E_m is computed experimentally, while clusters are searched for E_0 and E_1.

These techniques cannot apply to our case. First, the bit-oriented nature of the ciphers we are looking at makes us unable to use a two-step approach starting with a truncated analysis. Second, and as developed in the previous sections, the boomerang constraints we have are not probabilistic, so we cannot deal with the middle part as they do. These reasons lead us to the introduction of a new model.

5.2 Our New Approach

Our aim is to propose a model for quadratic Feistel ciphers that searches for boomerang distinguishers of various types (in particular related-key, related-key rotational-xor and related-key rotational-xor-differential) that satisfy the boomerang constraints in all the identified middle rounds.

General structure. The model takes as input three parameters: the number of rounds of E_0, of E_m and of E_1, hereafter denoted $|E_0|$, $|E_m|$ and $|E_1|$. In the case of Simon and since it has an influence on the characteristics that can be built for some attacks, the index of the round at which starts the distinguisher is also passed to the model. An overview of the distinguisher structure searched by the solver is presented in Figure 8. It is assumed that the characteristics in E_0 and E_1 are independent one from the other and that E_m captures all the dependencies. Differently stated, it is assumed that the characteristics that start from the top are uniform in E_1, and conversely, the characteristics that start from the bottom are uniform in E_0. This induces that the cost of these parts of the distinguisher are the square of their characteristics probability (the usual p^2 and q^2 terms (respectively) of the final probability). The characteristics over E_0 and E_1 are fully specified.

On the other hand, the second parameter $|E_m|$ is related to the part where the characteristics over E_0 and E_1 intermingle, so in which the boomerang constraints apply. Since in the case of quadratic Feistel ciphers the boomerang constraint is either verified or not and only depends on the (rotational-xor) differences in the states, we must force the boomerang constraint to be computable, and verified. This is a crucial difference with boomerang modelling for other primitives, where the round constraint is probabilistic.

Parts of the differences of the middle rounds come from the probability-one propagation of the differences of E_0 and E_1, while we leave open the possibility for the model to fix or not the difference in the other bits. We simply impose that all the boomerang constraints in these rounds have to be verified, and the model decides when to enforce some bit difference so that this objective is met. The probability of this part is denoted r, so the

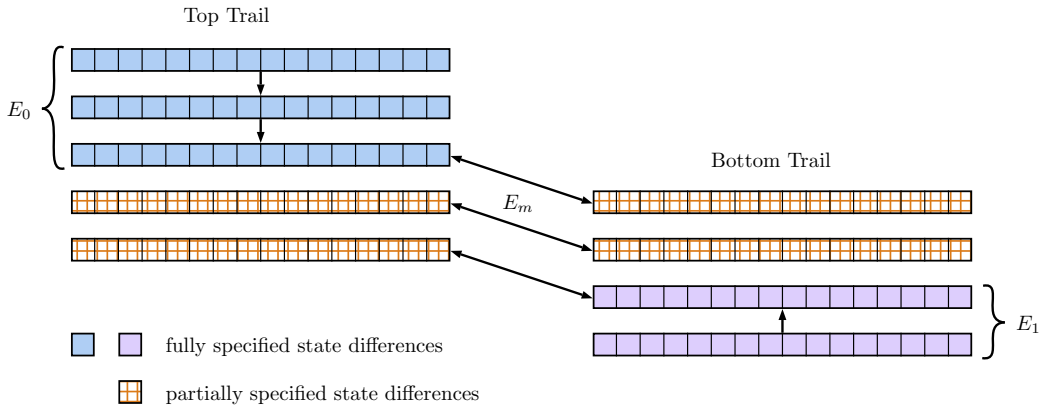

Figure 8: Organisation of the states in our SMT model. Example with $|E_0| = 2$, $|E_m| = 3$ and $|E_1| = 1$. The states corresponding to E_0 and to E_1 are fully determined. Their propagation probability is squared in the final cost formula. $|E_m|$ indicates the number of boomerang constraints that are verified, and determines partially specified states. The propagation probability in E_m is *not* squared in the final cost formula.

final probability of the distinguisher is p^2q^2r. An example of a solution returned by our model for 140 rounds of KATAN is presented in Appendix F, where the undetermined bit differences are represented with light colours. We detail below the exact rules used in our model.

Note that leaving some bits undetermined is equivalent to considering the cost of a cluster of possible states in the middle. In any case, we verified the validity of the model in general and of the cut (E_0, E_m, E_1) in particular for each of our run by confronting the returned probability estimate to the one experimentally obtained with tests programmed in C language.

Difference propagation. As our target ciphers are bit-oriented, our model also needs to be bit-oriented. Hence, we consider that each bit of difference has a value in $\{0, 1, ?\}$, where '?' represents an unknown difference. There are then simple propagation rules, as the only nonlinear operation we have to model is the AND. We model it according to Table 10: an inactive AND outputs an inactive bit, and if a AND is known to be active, then the output is uniform and it is possible to force its output with an extra-cost in probability of 1/2 (but we can also leave it undetermined), and if we don't know if the AND is active because of unknown input bit differences its output is unknown.

Table 10: Propagation rules and corresponding constraints for a single AND.

Input (δ_x, δ_y)	Output δ_{out}	Constraint on (x, y)
$(0, 0)$	0	None
$(1, 0)$	0 or 1 (proba 1/2) or ?	$y = \delta_{out}$
$(0, 1)$	0 or 1 (proba 1/2) or ?	$x = \delta_{out}$
$(1, 1)$	0 or 1 (proba 1/2) or ?	$x \oplus y = \delta_{out} \oplus 1$
$(1, ?), (?, 1)$	0 or 1 (proba 1/2) or ?	Unknown
$(0, ?), (?, 0), (?, ?)$	?	Unknown

These simple rules are sufficient to deal with KATAN32 as there are only 2 AND in each round, each dealing with different bits. But to handle more complex functions as the one of SIMON where all the input bits appear in several AND, we need to be careful. Many papers studied the differential behaviour of one round of SIMON-like ciphers, notably the article by Stefan Kölbl *et al.* [KLT15] in which a closed-form expression for the exact differential behaviour of one round is given:

Theorem 3 ([KLT15])**.** *Let $f(x) = x \lll a \cdot x \lll b \oplus x \lll c$ be an n-bit function, where $gcd(n, a-b) = 1$, n even, and $a > b$. Let α and β be an input and output difference, wt represent the Hamming weight and define*

$$\begin{aligned} \texttt{varibits} &= (\alpha \lll a) \vee (\alpha \lll b), \\ \texttt{doublebits} &= (\alpha \lll b) \cdot \overline{(\alpha \lll a)} \cdot (\alpha \lll (2a-b)) \\ \gamma &= \beta \oplus (\alpha \lll c). \end{aligned}$$

The probability that the input difference α leads to the output difference β is

$$Pr(\alpha \to \beta) = \begin{cases} 2^{-n+1} & \text{if } \alpha = 1 \text{ and } wt(\gamma) \equiv 0 \mod 2, \\ 2^{-\mathrm{wt}(\texttt{varibits}+\texttt{doublebits})} & \text{if } \alpha \neq 1 \text{ and } \gamma \cdot \overline{\texttt{varibits}} = 0 \\ & \text{and } (\gamma \oplus \gamma \lll (a-b)) \cdot \texttt{doublebits} = 0, \\ 0 & \text{else.} \end{cases} \tag{5}$$

As was proven in [LLA+20], the same equations can be used to deal with the case or RX differentials.

This formula is directly used when searching for the differential characteristics used in E_0 and E_1, and is slightly tweaked when dealing with E_m. Another way of looking at the differential properties is by considering the impact the choice of the output differential of an active AND has on the value of its input (which is shown in the last column of Table 10).

Besides the above result on SIMON, there has been other works that study differential propagation through an AND (or, more generally, through a quadratic function). For example, the Keccak reference [BDPA11] gives the exact propagation probabilities of differences through χ.

Overall, we have the following propagation rules, which fully capture the differential properties of quadratic functions:

Proposition 6 (Propagation rules for correlated AND)**.** *If multiple active AND share a common input bit:*

- *If this bit is inactive, all AND must have the same output difference*
- *If this bit is active, the output of the AND are independent, except:*
- *If there is a chain of input bits linked by an AND that forms a loop (as represented in Figure 9), the number of 0 output differences among the AND involved in the loop is even.*

Proof. The first case comes from the fact that fixing the output differential of an AND when the input differential is $(0,1)$ or $(1,0)$ fixes the value of the inactive input bit. The second and third case stem from the fact that with input differential $(1,1)$, the constraint is whether the value of the two input bits are equal (output 1) or different (output 0). This is always possible, except if there is a cycle, in which case there must be an even number of 0 among the output differential of the AND involved in the cycle, otherwise the cycle of equations would imply $x \neq x$. □

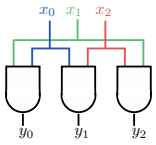

Figure 9: An AND loop. If all the x_i are active, 0 or 2 of the y_i are inactive.

We remark that it is possible to recover the nice closed formulas of Theorem 3 from Proposition 6. More generally, these rules allow to compute the propagation of a differential through any quadratic function. The only thing to note regarding our model is that the AND propagation rule allows to choose the result of any bit that contains an active AND, with some constraints if two bits come from AND that share one input: if there is an unknown difference involved in the computation of one of the two bits, we can only force the output difference of one of them, and if the shared bit is inactive, the output difference of the two AND must be equal.

Boomerang constraint. In addition to the equations constraining the difference propagation and keeping track of its cost, we add equations that force all the E_m middle rounds to verify the boomerang constraints. As we detailed in the previous sections, these boomerang constraints are quadratic expressions in the differences (for instance, for SIMON we need to verify $(\alpha_\ell \lll 8) \cdot (\delta_r \lll 1) \oplus (\alpha_\ell \lll 1) \cdot (\delta_r \lll 8) = 0$).

Implementations. We first implemented this model for KATAN32 in MILP, using Gurobi [GO21]. It was unfortunately very costly in time and memory, and we only managed to obtain optimal solutions in reasonable time (less than a week on our machine) for up to 140 rounds.

We then switched to an SMT modeling of the problem, which ended up being much more natural for bit-oriented ciphers (a detailed explanation of our model is provided in Appendix I). We used Bitwuzla [NP23] for our experiments, as it is at this time one of the fastest solvers for bitvector theories[5]. This has some drawbacks, as we do not optimize a model, but check its satisfiability. Thus, we need to run the solver with different target objectives to find a maximum or a minimum. As the solver parallelizes poorly, we decided to launch several instances with different parameters in parallel to compensate these effects and benefit from the several cores available on our machine.

Note that each run of the SMT solver took less than one day to solve, except for some instances of KATAN where it took two days. Experiments were performed on a computer with two AMD EPYC 7282 processors, for a total of 64 parallel threads at 3.2GHz. Our implementations of this model is available at https://github.com/xbonnetain/quadratic-feistel-boomerangs.

5.3 Application to KATAN32

As reported in Table 11, our model gave us distinguishers on KATAN32 and predicted their probability fairly accurately in general, the highest gap being less than 3 bits. These discrepancies might occur because of clustering effects or because of our restriction to parallel-side distinguishers.

[5]As reported on https://bitwuzla.github.io/, Bitwuzla won 26 out of 56 (participated) division awards at SMT-COMP 2023.

Table 11: Results of our SMT model for KATAN32.

Rounds	140	141	142	143	144	145	146	147	148	149	150	151	152
Top	40	40	41	41	42	40	43	43	44	45	45	45	46
Middle	60	61	60	61	60	65	60	61	60	59	60	61	60
Bottom	40	40	41	41	42	40	43	43	44	45	45	45	46
Model proba.$^\diamond$	-17	-17	-20	-21	-22	-23	-25	-25	-27	-29	-31	-31	-33
Time*	10min	1h30	4h40	6h	7h	4h30	10h	7h	12h	29h	43h	41h	50h
Exp. proba.$^\diamond$	-16.2	-15.8	-20.2	-19.8	-19.1	-22.5	-24.3	-24.3	-26.3	-28.3	-30.2	-30.1	-31.7

* The reported time is the highest between the time to find the best set of characteristics and the time to prove no better set exist.

$\diamond$ Binary logarithm of the probabilities.

5.4 Application to Simon-32/64

We searched for distinguishers on Simon-32/64 with different cuts. We did not see much of a difference, except that with more middle rounds, the model becomes harder to solve.

5.4.1 RX-Boomerangs

Our model for Simon builds upon the rotational-xor SMT model from [LLA+20]. As all round operations commute with a bitshift, the differential properties of Simon are the same as its rotational-xor properties. Thus, we can reuse the differential model. This model does not correctly propagate the case where the input of the round function is all-active, but as this would imply a very low-probability transition, this is unlikely to occur. We generalize their model to support incomplete characteristics with unknown differences, and added key relations and equations for the boomerang constraints in the middle. The results are summarized in Table 12. We looked for distinguishers with many starting rounds, and round 3 was generally an optimal choice.

Table 12: Results of our SMT model for RX-boomerangs on Simon-32/64.

Rounds	13	14	15	16	17	18	19
Starting round	3	3	10	3	3	3	3
Cut	4+5+4	5+4+5	5+5+5	5+6+5	6+5+6	6+6+6	7+5+7
Model proba.$^\diamond$	0	-3	-6	-12	-16	-24	-30
Experimental proba.$^\diamond$	0	-3	-6	-12	-16	-24	-29.5

$\diamond$ Binary logarithm of the probabilities.

We experimentally checked the distinguishers returned by the SMT solver to be sure that the model output matches the actual probability. Note that another possibility would have been to use the technique shown in [SRB21] by Sadeghi *et al.* and adapted later by Lu *et al.* in [LLA+22] to include in the model a verification that at least one quartet follows the boomerang characteristic. Such technique would make sure the probability is non-zero, but it does not help to correctly estimate the actual probability of the boomerang.

5.4.2 RX-Differential Boomerangs

We tweaked our previous model to support RX-differential boomerangs as defined in [CZX+23]. The differences are as follows:

- the lower characteristics are single-key,
- as detailed in Section 3.3, the bottom difference used in the boomerang constraint is rotated.

Our results are presented in Table 13. As in [CZX+23] a distinguisher for 16-round Simon-32/64 with experimental probability $2^{-31.89}$ was proposed, we looked for a 16-round distinguisher with theoretical probability at least 2^{-32}, but failed to find one. The boomerang of highest probability we could find had a theoretical probability of 2^{-36}, and as expected, it was not experimentally confirmed.

Table 13: RX-differential distinguishers for Simon-32/64.

Rounds	13	14	15
Starting round	3	3	3
Cut	7+4+2	5+5+4	6+5+4
Model proba.$^\diamond$	-17	-23	-28
Experimental proba.$^\diamond$	-17.2	-21	-27.6

$\diamond$ Binary logarithm of the probabilities.

5.4.3 Standard Boomerangs

We adapted our RX model to standard boomerangs, the only change being that we removed the rotational difference that stemmed from the constants in the key schedule.

An important difference with rotational-xor boomerangs for Simon is that these distinguishers are independent of the starting round. Our findings are summarized in Table 14.

Table 14: Related-key boomerang distinguishers for Simon-32/64.

Rounds	12	13	14	15	16	17
Cut	5+2+5	5+3+5	5+4+5	6+3+6	5+6+5	6+5+6
Model proba.$^\diamond$	0	-3	-7	-11	-19	-25
Experimental proba.$^\diamond$	0	-2.7	-6.7	-10.4	-18.8	-23.6

$\diamond$ Binary logarithm of the probabilities.

6 Rotational-Xor Rectangle Attacks on Simon-32/64

In this section, we propose two attacks based on our two best distinguishers on Simon-32/64. Note that several distinguishers are optimal for 18 and 19 rounds, so we picked the ones with the smaller number of active bits in α and δ.

6.1 Related-Key Rectangle Key-Recovery Procedure

Before giving the details of our attacks, we start by introducing our key-recovery technique, which is a variant of [ZDM+20].

As can be seen on Figure 10, the key-recovery rounds that are added before and after the distinguisher rounds (E_d) are denoted by E_b and E_f respectively. An important parameter to determine the cost of the key-recovery process is the number of bits that are unknown in the plaintext difference (denoted r_b) and in the ciphertext difference (denoted r_f) when assuming that a quartet is valid. These are computed by propagating with probability 1 the difference at the beginning of the distinguisher (α) up to the plaintexts for r_b, and similarly by propagating with probability 1 the difference δ down to the ciphertexts for r_f.

In addition to this, another important parameter is the number of key bits the attacker needs to guess in the procedure. Let m_b be the number of bits of key from E_b one needs to know to generate the pairs from the plaintexts, and m_f be the number of bits of key from E_f required to compute the collision function leading to the quartets from the pairs. We denote by $\mathbf{f}$ the number of filter bits, that is, bits on which we collide the pairs to construct a candidate quartet. Finally, we denote by m_q the number of additional bits one needs to know to check that the difference is δ.

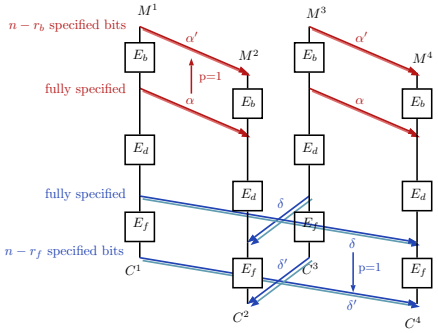

Figure 10: Rotational-xor rectangle key-recovery setup.

Data complexity. We consider that an attacker wants to obtain s quartets that follow the boomerang distinguisher of probability P. The data pool is organized with structures of messages, which correspond to sets of plaintexts taking all the possible values over the r_b unknown bits while having a fixed and common value where the difference is known. Such a first set is encrypted under K_1. The set containing the M^2 messages that could potentially be paired with them is computed by taking all the messages of the first set (or their rotated version in case of an RX variant) and adding the bits of the difference that are known for sure. It is thus encrypted under K_2. A similar procedure is followed to build the messages encrypted under K_3 and K_4.

A total of $y = \sqrt{\frac{s}{P}} \times 2^{\frac{n}{2}-r_b}$ structures encrypted under each of the 4 related keys are required to get s right quartets, so the data complexity of the attack is $D = 4\sqrt{\frac{s}{P}} \times 2^{\frac{n}{2}}$.

Attack procedure. The key-recovery process is as follows:

1. Construct y structures of 2^{r_b} messages each under the 4 related keys K_1, K_2, K_3, K_4 and get four plaintext/ciphertext sets $L^i = (M^i, C^i)$ for $(i = 1, \cdots, 4)$. Sort L^2 and L^4 according to the r_b bits of plaintexts (for a faster access in the next step).

2. For each of the 2^{m_b} possibilities for the relevant key bits of E_b:

   (a) For each structure and for each $M^1 \in L^1$, use the m_b bits of guess to compute M^2 from M^1. Fetch M^2 in L^2 and build the set

   $$S^1 = \{(M^1, C^1, M^2, C^2) | (M^1, C^1) \in L^1, (M^2, C^2) \in L^2, E_b(M^1) \oplus E_b(M^2) = \alpha\}.$$

Similarly, build

$$S^2 = \{(M^3, C^3, M^4, C^4) | (M^3, C^3) \in L^3, (M^4, C^4) \in L^4, E_b(M^3) \oplus E_b(M^4) = \alpha\}.$$

(b) For each of the 2^{m_f} possibilities for the relevant key bits of E_f:

i. Initialize a list of 2^{m_q} counters to track the number of hits of each of the guesses.

ii. Using bits of m_f, compute the $\mathbf{f}$ filter bits to collide on and sort S^1 and S^2 according to them.

iii. Look for a collision between these values in S^1 and S^2. The number of quartets obtained is equal to $y^2 \times 2^{2r_b} \times 2^{-2\mathbf{f}}$.

iv. Use these quartets to deduce information on the m_q key bits. Use a guess-and-filter process to guess one bit at a time and reduce the number of conforming quartets. This time complexity is denoted by ε.

The time complexity of the full procedure is defined by the most expensive steps, that are step 2.(a) of time complexity equal to $2^{m_b+r_b} \times y \times 4 \times \frac{|E_b|}{|E_b|+|E_d|+|E_f|}$, step 2.(b).ii of time complexity equal to $2^{m_b+m_f+r_b} \times y \times 4 \times \frac{\rho}{|E_b|+|E_d|+|E_f|}$, with ρ the number of rounds one needs to process to compute the filter bits, step 2.(b).iii of time complexity $2^{m_b+m_f+r_b} \times y \times 2 \times C$, with C the cost to insert a pair to find a collision and by step 2.(b).iv of time complexity equal to $\frac{s}{P} \times 2^{m_b+m_f+n-2\mathbf{f}} \times \varepsilon$ (for each remaining quartet, some key guesses are made and used to check the characteristic in E_f).

Estimation of ε. As we have a bit-oriented cipher, we can guess one bit at a time. We observe that we can generally obtain 1 bit of filter from 1 guessed key bit, if the activity pattern of the AND is known. If this is not the case, we would need 2 key guesses, but only once, as the knowledge of these two bits allows to invert the other AND in which these bits are involved at the cost of 1 key guess. Moreover, each bit of filter divides the number of quartets by 4, as we have a condition on two pairs. Hence, with Q quartets, we can estimate the cost of eliminating the wrong quartets as

$$2 \times (Q + 2 \times (Q/4 + 2 \times (Q/16 \dots))) \simeq 4Q.$$

Now, the cost of propagating the key guess can be conservatively estimated as 1 round of Simon for each ciphertext in the quartet. Hence, unless stated otherwise, we will consider that $\varepsilon = 16/r$, with r the number of rounds.

Estimation of C. We need to estimate the cost per element of collision search relative to one encryption. This is heavily dependent on the architecture and the target cipher. For the sake of simplicity, we choose $C = 1$.

Preprocessing. As the application of the round function f in the first and in the last round of Simon does not involve the key, we first preprocess our data to remove these two f-applications. This part will always have a negligible cost in our case.

6.2 Key guesses and propagation of differences

We need to guess key bits in order to be able to deterministically form pairs of plaintexts ((M^1, M^2) and (M^3, M^4)) and also to determine the difference pattern at the end (to have bits to filter on). We use the following principles:

1. To know the output difference of an AND, it suffices to know:

(a) Nothing if both input bits are inactive,

(b) The value of an inactive input bit if there is one inactive bit and one active bit,

(c) Whether the two inputs are equal if both bits are known to be active.

2. To know, up to a constant, the output difference of an AND, it suffices to know:

(a) The value of an inactive bit if the other bit is known up to a constant.

3. To know the output value of an AND, we need to know:

(a) One input value if this value is 0,

(b) The two inputs otherwise.

The first item comes from Proposition 6, the last one is straightforward, so we only need to prove the second item. We consider the difference between $x_0 \cdot (x_1 \oplus c_1)$ and $x_0 \cdot (x'_1 \oplus c'_1)$, assuming we know x_1, x'_1 and $c_1 \oplus c'_1$. Then the output difference is $x_0 \cdot x_1 \oplus x_0 \cdot x'_1 \oplus x_0 \cdot (c_1 \oplus c'_1)$. This means that in SIMON, we can filter with the output difference of 2 AND solely from the value of one inactive bit, even if we do not know the activity pattern of the other input bit.

6.3 Attack on 24 Rounds of Simon-32/64

In this first attack, we follow the technique described above and turn our 18-round distinguisher (depicted in Appendix G) into an attack of 24-round SIMON-32/64.

Our 18-round distinguisher is of probability $P = 2^{-24}$ and starts at round 3. We decide to use $|E_b| = 3$ rounds of key-recovery before it[6], and $|E_f| = 3$ rounds after it, and we aim at having $s = 4$ right quartets. By propagating with probability 1 the α and δ differences, we obtain that $r_b = 24$ and that $r_f = 21$. The computation of these two parameters is illustrated in Figure 11 and Figure 12. As noted above, we can remove the first and last round (except the round key addition). There are 3 inactive bits and 5 active bits in α', while there are 3 inactive bits and 6 active bits in δ'. Two additional relations are known in δ', between two bit differences computed from two active AND implying the same inactive bits (see the bits circled in red and blue in Figure 12), which, from Proposition 6 results in the same activity after the AND.

To fully compute the difference in E_b, we need to know the values of 8 bits in x^2_ℓ. Hence, we have to guess 8 key bits. Then, we need to propagate these differences from round 2 to round 1. We remark that one bit of x^1_ℓ, the bit number 2, is not needed to compute the required values in x^2_ℓ or to propagate the differences, as it is combined with two inactive bits in the round function (principle 1.(b)). Hence, only 15 additional bits are needed to generate the pairs, for a total of $m_b = 23$ bits. As can be seen in Figure 12, $\mathbf{f} = 11$ bits of filter are directly accessible in the ciphertexts. Hence $\rho = 0$ for this attack as we do not decrypt to compute the filter bits.

Following the formula recalled previously, the number of structures required to conduct the attack is equal to $y = \sqrt{\frac{4}{2^{-24}}} \cdot 2^{\frac{32}{2}-24} = 2^5$, so the final data complexity is equal to $4 \times 2^5 \times 2^{24} = 2^{31}$. Using $C = 1$ and $\varepsilon = 16/24$, we obtain that step 2.(a) is of complexity $2^{50.4}$, there is no bit in m_f so step 2.(b).ii is free, step 2.(b).iii is of complexity 2^{53} and step 2.(b).iv is of complexity $16/24 \times 4 \times 2^{23+24} \times 2^{32-22} = 2^{58.4}$. Overall, this costs $2^{58.5}$, and the cost of recovering the remaining master key bits is negligible compared to that of the previous steps.

[6] Thus we attack the 24 first rounds.

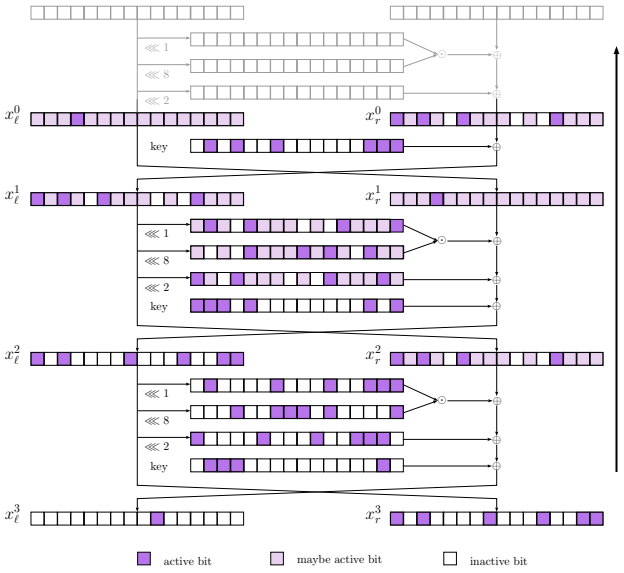

Figure 11: E_b for our 24-round attack.

6.3.1 Better Filtering

As the cost of processing the quartets was the bottleneck, we can improve the attack by guessing more key bits, assuming they give us more filtering bits. We can guess one bit to compute the value of the bit 0 of x_ℓ^{22}, which is inactive. Knowing this value gives us 2 additional bits of filter, from principle 2.(a). The same thing occurs with bit 12 of x_ℓ^{22}. Overall, we have $\mathbf{f} = 15$ bits of filter with 2 more key guesses. In this case, $\rho = 1$, as we partially decrypt 1 round in E_f.

Hence, the cost of step 2.(b).iii becomes 2^{55} and the cost of step 2.(b).iv becomes $2^{52.4}$. Moreover, step 2.(b).ii is no longer free, and costs $2^{51.4}$. Overall, the attack costs $2^{55.4}$.

6.3.2 Optimizing Key Guesses

In the attack, we generate the pairs by guessing 23 bits. However, a careful study shows that some bits are not always needed, which means there are some pairs we do not need to reconstruct for some key guesses. We cannot gain anything in x_ℓ^2, but there are some possible optimizations in x_ℓ^1. In more details:

- We need to know the value of bit 4 of x_ℓ^1 only if bit 13 of x_ℓ^1 is active (principle 1.(b)),
- We need to know the value of bit 0 of x_ℓ^1 only if bit 7 of x_ℓ^1 is active (principle 1.(b), to propagate a difference) or if bit 9 of x_ℓ^1 is 1 (principle 3.(a), to compute a value in x_ℓ^2),
- We need to know the value of bit 13 of x_ℓ^1 only if bit 6 of x_ℓ^1 is active (principle 1.(b), to propagate a difference) or if bit 6 of x_ℓ^1 is 1 (principle 3.(a), to compute a value in x_ℓ^2).

These 3 events are independent, and each constraint occurs with probability 1/2. Hence, we need to guess the key bit masking bit 4 half the time and the key bits masking bits 0 and 13 3/4 of the time. Overall, instead of multiplying the cost by 8, we multiply it by $\left(\frac{1}{2} + \frac{2}{2}\right)\left(\frac{1}{4} + \frac{2\times 3}{4}\right)^2 \simeq 4.6 \simeq 2^{2.2}$. This improvement is only for the pairs, which reduces the cost of step 2.(a), step 2.(b).ii and step 2.(b).iii by a factor $2^{3-2.2} = 2^{0.8}$. For the

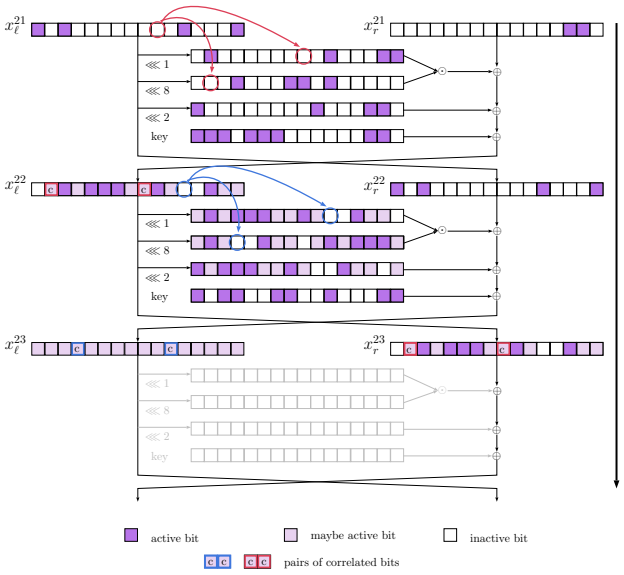

Figure 12: E_f for our 24-round attack.

quartets, we don't have to reprocess a quartet only if both pairs don't need the key guess. Hence, the cost becomes $\left(\frac{1}{4}+\frac{6}{8}\right)\left(\frac{1}{16}+\frac{30}{16}\right)^2 \simeq 6.6 \simeq 2^{2.7}$. The cost of step 2.(b).iv is reduced by a factor $2^{3-2.7} = 2^{0.3}$.

Overall the cost is $2^{54.6}$.

6.4 Attack on 25 Rounds of Simon-32/64

The attack on 25 rounds follows the same techniques as the 24-round attack above, but with our 19-round distinguisher (depicted in Appendix H). As the input differential and the top key relations are the same for the 18-round and the 19-round distinguisher, we can reuse part of the previous attack. We still add 3 rounds before and after, and the probability of the distinguisher is $P = 2^{-30}$. As we want 4 right quartets, we take the full codebook over the 4 related keys, which is a data complexity of 2^{34}.

The bottom part of the key-recovery is detailed in Figure 13. There are 5 active bits, 3 inactive bits and 2 correlated bits in δ', which are 9 bits of filter for free. We can guess the key masking the 3 inactive bits in x_ℓ^{23} to gain 6 additional bits of filter, from principle 2.(a). Finally, there are two AND with 2 active inputs, which means we can get 2 more bits of filtering from 2 key guesses, from principle 1.(c). To balance the costs, we guess all 3 key bits masking an inactive bit, for a total of $\mathbf{f} = 15$ bits of filter.

With the previous optimizations, step 2.(a) costs $\frac{2}{25}2^{22.2+32+2} \simeq 2^{52.6}$. Step 2.(b).ii costs $\frac{1}{25}2^{22.2+3+32+2} \simeq 2^{54.6}$. Step 2.(b).iii costs $2^{22.2+3+32+1} = 2^{58.2}$. Step 2.(b).iv costs $\frac{16}{25}2^{22.7+4+32+32-2\times15} \simeq 2^{59.1}$. Overall the cost is $2^{59.7}$.

Conclusion

Our research provides new evidences that the naive construction of boomerang distinguishers is inappropriate, and demonstrates once again that the top and bottom characteristics cannot be considered independent. The specific case of quadratic Feistel ciphers is studied in detail and it is shown that the boomerang condition has a simple expression, which makes

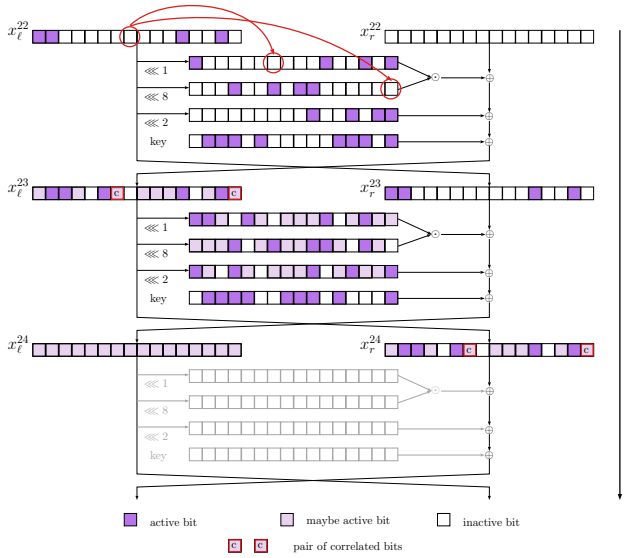

Figure 13: E_f for our 25-round attack.

a connection either feasible or not, independently of the state values. Several published distinguishers on KATAN and SIMON are proved wrong, either theoretically or experimentally. A new SMT-solver-based tool is proposed to automatically find boomerangs that take into account the boomerang constraints in KATAN and SIMON. Notably, we obtain on SIMON-32/64 a 13-round distinguisher of probability 1 and a 25-round attack.

Acknowledgments

The authors would like to thank Schloss Dagstuhl and the organizers of the Dagstuhl Seminar 22141 "Symmetric Cryptography" where this work was initiated, and Yu Sasaki for suggesting the study of this topic.

This work has been partially supported by the French Agence Nationale de la Recherche through the DeCrypt project under Contract ANR-18-CE39-0007 and the OREO project under Contract ANR-22-CE39-0015.

References

[AL16] Tomer Ashur and Yunwen Liu. Rotational cryptanalysis in the presence of constants. *IACR Trans. Symm. Cryptol.*, 2016(1):57–70, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/535.

[ALLW13] Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential and linear cryptanalysis of reduced-round Simon. Cryptology ePrint Archive, Report 2013/526, 2013. https://eprint.iacr.org/2013/526.

[ALLW15] Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential cryptanalysis of round-reduced Simon and Speck. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *LNCS*, pages 525–545. Springer, Heidelberg, March 2015.

[BDK01] Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack - rectangling the Serpent. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 340–357. Springer, Heidelberg, May 2001.

[BDK05] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 507–525. Springer, Heidelberg, May 2005.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak reference, 2011. https://keccak.team/files/Keccak-reference-3.0.pdf.

[BHL+20] Hamid Boukerrou, Paul Huynh, Virginie Lallemand, Bimal Mandal, and Marine Minier. On the Feistel counterpart of the boomerang connectivity table (long paper). *IACR Trans. Symm. Cryptol.*, 2020(1):331–362, 2020.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, Heidelberg, December 2009.

[BR11] Andrey Bogdanov and Christian Rechberger. A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *LNCS*, pages 229–240. Springer, Heidelberg, August 2011.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 2–21. Springer, Heidelberg, August 1991.

[BSS+15] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. SIMON and SPECK: Block ciphers for the internet of things. Cryptology ePrint Archive, Report 2015/585, 2015. https://eprint.iacr.org/2015/585.

[CCW+18] Zhihui Chu, Huaifeng Chen, Xiaoyun Wang, Xiaoyang Dong, and Lu Li. Improved integral attacks on SIMON32 and SIMON48 with dynamic key-guessing techniques. *Secur. Commun. Networks*, 2018:5160237:1–5160237:11, 2018.

[CHP+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 683–714. Springer, Heidelberg, April / May 2018.

[CTS+16] Jiageng Chen, Jesen Teh, Chunhua Su, Azman Samsudin, and Junbin Fang. Improved (related-key) attacks on round-reduced KATAN-32/48/64 based on the extended boomerang framework. In Joseph K. Liu and Ron Steinfeld, editors, *ACISP 16, Part II*, volume 9723 of *LNCS*, pages 333–346. Springer, Heidelberg, July 2016.

[CW16] Huaifeng Chen and Xiaoyun Wang. Improved linear hull attack on round-reduced simon with dynamic key-guessing techniques. In Thomas Peyrin, editor, *FSE 2016*, volume 9783 of *LNCS*, pages 428–449. Springer, Heidelberg, March 2016.

[CZX+23] Siwei Chen, Mingming Zhu, Zejun Xiang, Runqing Xu, Xiangyong Zeng, and Shasha Zhang. Rotational-xor differential rectangle cryptanalysis on simon-like ciphers. In Mike Rosulek, editor, *Topics in Cryptology - CT-RSA 2023 - Cryptographers' Track at the RSA Conference 2023, San Francisco, CA, USA, April 24-27, 2023, Proceedings*, volume 13871 of *Lecture Notes in Computer Science*, pages 305–330. Springer, 2023.

[DDK09] Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *CHES 2009*, volume 5747 of *LNCS*, pages 272–288. Springer, Heidelberg, September 2009.

[DDV20] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. Catching the fastest boomerangs application to SKINNY. *IACR Trans. Symm. Cryptol.*, 2020(4):104–129, 2020.

[DF16] Patrick Derbez and Pierre-Alain Fouque. Automatic search of meet-in-the-middle and impossible differential attacks. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 157–184. Springer, Heidelberg, August 2016.

[DKS10] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 393–410. Springer, Heidelberg, August 2010.

[GO21] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[HBS21] Hosein Hadipour, Nasour Bagheri, and Ling Song. Improved rectangle attacks on SKINNY and CRAFT. *IACR Trans. Symm. Cryptol.*, 2021(2):140–198, 2021.

[HKLP05] Seokhie Hong, Jongsung Kim, Sangjin Lee, and Bart Preneel. Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In Henri Gilbert and Helena Handschuh, editors, *FSE 2005*, volume 3557 of *LNCS*, pages 368–383. Springer, Heidelberg, February 2005.

[HNE22] Hosein Hadipour, Marcel Nageler, and Maria Eichlseder. Throwing boomerangs into Feistel structures application to CLEFIA, WARP, LBlock, LBlock-s and TWINE. *IACR Trans. Symm. Cryptol.*, 2022(3):271–302, 2022.

[ISC13] Takanori Isobe, Yu Sasaki, and Jiageng Chen. Related-key boomerang attacks on KATAN32/48/64. In Colin Boyd and Leonie Simpson, editors, *ACISP 13*, volume 7959 of *LNCS*, pages 268–285. Springer, Heidelberg, July 2013.

[JRS22] Amit Jana, Mostafizar Rahman, and Dhiman Saha. DEEPAND: In-depth modeling of correlated AND gates for NLFSR-based lightweight block ciphers. Cryptology ePrint Archive, Report 2022/1123, 2022. https://eprint.iacr.org/2022/1123.

[Kir15] Aleksandar Kircanski. Analysis of boomerang differential trails via a SAT-based constraint solver URSA. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *ACNS 15*, volume 9092 of *LNCS*, pages 331–349. Springer, Heidelberg, June 2015.

[KJK20] Bonwook Koo, Younghoon Jung, and Woo-Hwan Kim. Rotational-xor rectangle cryptanalysis on round-reduced simon. *Secur. Commun. Networks*, 2020:5968584:1–5968584:12, 2020.

[KKS01] John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and Serpent. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 75–93. Springer, Heidelberg, April 2001.

[KLT15] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON block cipher family. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 161–185. Springer, Heidelberg, August 2015.

[KN10] Dmitry Khovratovich and Ivica Nikolic. Rotational cryptanalysis of ARX. In Seokhie Hong and Tetsu Iwata, editors, *FSE 2010*, volume 6147 of *LNCS*, pages 333–346. Springer, Heidelberg, February 2010.

[LLA+20] Jinyu Lu, Yunwen Liu, Tomer Ashur, Bing Sun, and Chao Li. Rotational-XOR cryptanalysis of simon-like block ciphers. In Joseph K. Liu and Hui Cui, editors, *ACISP 20*, volume 12248 of *LNCS*, pages 105–124. Springer, Heidelberg, November / December 2020.

[LLA+22] Jinyu Lu, Yunwen Liu, Tomer Ashur, Bing Sun, and Chao Li. Improved rotational-xor cryptanalysis of simon-like block ciphers. *IET Inf. Secur.*, 16(4):282–300, 2022.

[LLW17] Zhengbin Liu, Yongqiang Li, and Mingsheng Wang. Optimal differential trails in SIMON-like ciphers. *IACR Trans. Symm. Cryptol.*, 2017(1):358–379, 2017.

[LMR22] Virginie Lallemand, Marine Minier, and Loïc Rouquette. Automatic search of rectangle attacks on Feistel ciphers: Application to WARP. *IACR Trans. Symm. Cryptol.*, 2022(2):113–140, 2022.

[LPS21] Gaëtan Leurent, Clara Pernot, and André Schrottenloher. Clustering effect in simon and simeck. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 272–302. Springer, Heidelberg, December 2021.

[Mur11] Sean Murphy. The return of the cryptographic boomerang. *IEEE Trans. Inf. Theory*, 57(4):2517–2521, 2011.

[NP23] Aina Niemetz and Mathias Preiner. Bitwuzla. In Constantin Enea and Akash Lal, editors, *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part II*, volume 13965 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2023.

[QHS15] Kexin Qiao, Lei Hu, and Siwei Sun. Differential security evaluation of simeck with dynamic key-guessing techniques. Cryptology ePrint Archive, Report 2015/902, 2015. https://eprint.iacr.org/2015/902.

[RG18] Raghvendra Rohit and Guang Gong. Correlated sequence attack on reduced-round Simon-32/64 and Simeck-32/64. Cryptology ePrint Archive, Report 2018/699, 2018. https://eprint.iacr.org/2018/699.

[RR16] Shahram Rasoolzadeh and Håvard Raddum. Improved multi-dimensional meet-in-the-middle cryptanalysis of KATAN. Cryptology ePrint Archive, Report 2016/077, 2016. https://eprint.iacr.org/2016/077.

[SFW15] Ling Sun, Kai Fu, and Meiqin Wang. Improved zero-correlation cryptanalysis on SIMON. In Dongdai Lin, XiaoFeng Wang, and Moti Yung, editors, *Information Security and Cryptology - 11th International Conference, Inscrypt 2015, Beijing, China, November 1-3, 2015, Revised Selected Papers*, volume 9589 of *Lecture Notes in Computer Science*, pages 125–143. Springer, 2015.

[SQH19] Ling Song, Xianrui Qin, and Lei Hu. Boomerang connectivity table revisited. *IACR Trans. Symm. Cryptol.*, 2019(1):118–141, 2019.

[SRB21] Sadegh Sadeghi, Vincent Rijmen, and Nasour Bagheri. Proposing an milp-based method for the experimental verification of difference-based trails: application to speck, SIMECK. *Des. Codes Cryptogr.*, 89(9):2113–2155, 2021.

[Wag99] David Wagner. The boomerang attack. In Lars R. Knudsen, editor, *FSE'99*, volume 1636 of *LNCS*, pages 156–170. Springer, Heidelberg, March 1999.

[WH19] Hongjun Wu and Tao Huang. Tinyjambu: A family of lightweight authenticated encryption algorithms. *Submission to the NIST Lightweight Cryptography Standardization Process (March 2019)*, 2019.

[WP19] Haoyang Wang and Thomas Peyrin. Boomerang switch in multiple rounds. *IACR Trans. Symm. Cryptol.*, 2019(1):142–169, 2019.

[YSS+22] Qianqian Yang, Ling Song, Siwei Sun, Danping Shi, and Lei Hu. New properties of the double boomerang connectivity table. *IACR Trans. Symm. Cryptol.*, 2022(4):208–242, 2022.

[ZDM+20] Boxin Zhao, Xiaoyang Dong, Willi Meier, Keting Jia, and Gaoli Wang. Generalized related-key rectangle attacks on block ciphers with linear key schedule: applications to SKINNY and GIFT. *Des. Codes Cryptogr.*, 88(6):1103–1126, 2020.

A A General Notion of Feistel Cipher

The aim of this section is to characterize the scope of this work, and identify on which constructions the previous result applies. First, we note that the only relevant part is to have a quadratic operation of the form $(x, y) \to (x, f(x) + y)$, that is, a quadratic function that does not affect its input bits. This works independently of the size of the branches. Hence, it also applies to generalized Feistel ciphers and NLFSR (which can be seen as contracting generalized Feistel ciphers with 1-bit branches). Second, the property is invariant by affine equivalence, meaning it also affects Lai-Massey schemes.

Looking for affected constructions, we obtained the following amusing result, that we failed to find in the literature. It tells us, that, in a sense, the Feistel construction is the only way to make a permutation from a function. This makes constructions affine-equivalent to a Feistel likely to be the only ones for which the Feistel boomerang constraint is relevant.

Lemma 1. *Let $R[h]$ be a round function that consists of affine functions and a call to a non-linear function h. If for all h, $R[h]$ is invertible, then $R[h]$ is affine-equivalent to a round of Feistel.*

Proof. We have that $R[h]$ has the form

$$R[h](x) = A_1(x) \oplus A_2 \circ h \circ A_3(x),$$

with A_i some affine functions, as it contains only one call[7] to h. As we consider affine-equivalence, we can normalize the input and output to restrict ourselves to linear functions.

Thus, we consider

$$R_2[h](x) = L_1(x) \oplus L_2 \circ h \circ L_3(x),$$

with L_i some linear functions.

If L_2 is all-zero, then $R_2[h]$ is linear, which is affine-equivalent to a degenerate Feistel-round with an all-zero inner function. If L_1 has not full rank, then it is possible to choose h such that the image of $L_2 \circ h$ does not contain any y in a supplementary of the image of L_1, thus $R_2[h](x)$ is not always invertible. Let's assume L_1 has full rank and L_2 is not all-zero. Then, up to affine equivalence the round function has the form

$$R_3[h](x) = x \oplus L_2 \circ h \circ L_3(x).$$

If L_3 has full rank, then the round function is equivalent to $x \oplus L_2 \circ g(x)$. As L_2 is not all-zero, there exists a g that maps 0 to 0 and a non-zero value in the image of L_2 to one of its preimages. This forces a collision, making the round non-injective.

Thus, let's assume L_3 has lower rank. Hence, we can decompose the input x as (a, b), with a in the kernel of L_3 and b in a supplementary space. Hence, we can split the input and output in two and write the round function as

$$R_4[h](a, b) = (a \oplus M_1 \circ h_1 \circ M_3(b), b \oplus M_2 \circ h_2 \circ M_3(b)),$$

with M_3 a linear permutation corresponding to L_3, and M_1, M_2 being the two components of L_2. Now, we remark that M_3 has a full rank, meaning that by the argument above, the function $g(b) = b \oplus M_2 \circ h_2 \circ M_3(b)$ can have collisions if M_2 is non-zero. From b_1, b_2 such that $g(b_1) = g(b_2)$, we can directly compute many a_1 and a_2 such that $R[h](a_1, b_1) = R[h](a_2, b_2)$, making the round non-invertible. Hence, M_2 is all-zero. In the end, we have that our round function is affine equivalent to

$$R_5[h](a, b) = (b, a \oplus M_1 \circ h_1(b)),$$

which is a Feistel round with the inner function $M_1 \circ h_1$.

□

B Proof of Theorem 2

Proof. For (M^3, M^4) to be an RX pair of RXD α, the following relation has to be satisfied (we use the same notations as in Figure 5):

$$M^4 = \overrightarrow{M^3} \oplus \alpha.$$

By expressing this with respect to M^1 only we get:

$$\begin{aligned}
M^4 &= \overrightarrow{M^3} \oplus \alpha \\
\iff E^{-1}(C^4) &= \overrightarrow{E^{-1}(C^3)} \oplus \alpha \\
\iff E^{-1}(\overrightarrow{C^1} \oplus \delta) &= \overrightarrow{E^{-1}(\overleftarrow{C^2 \oplus \delta})} \oplus \alpha \\
\iff E^{-1}(\overrightarrow{E(M^1)} \oplus \delta) &= \overrightarrow{E^{-1}(\overleftarrow{E(M^2) \oplus \delta})} \oplus \alpha \\
\iff E^{-1}(\overrightarrow{E(M^1)} \oplus \delta) &= \overrightarrow{E^{-1}(\overleftarrow{E(\overrightarrow{M^1} \oplus \alpha) \oplus \delta})} \oplus \alpha
\end{aligned}$$

□

[7] Note that the $R[h]$ are not the functions extended affine-equivalent to h, as the affine functions do not need to be invertible.

C Details on the Incompatibilities for KATAN

Table 15: Bit differences that are determined with probability 1 knowing that the difference in round 54 is ε (forward propagation).

rd			α_3				α_8		α_{10}		α_{12}							α'_5		α'_8					k_a	k_b	IR
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
60	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
61	1	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
62	1	1	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
63	?	1	1	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
64	0	?	1	1	?	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0	0	1	0	0	0
65	0	0	?	1	1	?	0	0	0	0	0	0	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0	1	0	0	1
66	0	0	0	?	1	1	?	0	0	0	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0	0	0	1	0	1	0
67	1	0	0	0	?	1	1	?	0	0	0	0	0	0	0	0	0	?	?	?	?	0	0	0	0	0	0	0	0	0	0
68	0	1	0	0	0	?	1	1	?	0	0	0	0	0	0	0	0	?	?	?	?	?	0	0	0	0	0	0	0	0	0
69	0	0	1	0	0	0	?	1	1	?	0	0	0	0	0	0	0	?	?	?	?	?	?	0	0	0	0	0	0	0	0
70	?	0	0	1	0	0	0	?	1	1	?	0	0	0	0	0	0	0	?	?	?	?	?	?	0	0	0	0	0	0	0
71	?	?	0	0	1	0	0	0	?	1	1	?	0	0	0	0	0	0	?	?	?	?	?	?	?	0	0	0	0	0	1
72	?	?	?	0	0	1	0	0	0	?	1	1	?	0	0	0	0	0	?	?	?	?	?	?	?	?	0	0	0	0	1
73	?	?	?	?	0	0	1	0	0	0	?	1	1	?	0	0	0	0	0	?	?	?	?	?	?	?	?	?	0	0	0	1	0	1
74	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	0	0	0	0	?	?	?	?	?	?	?	?	?	?	0	0	0	1
75	?	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	0	0	0	?	?	?	?	?	?	?	?	?	?	?	0	0	1
76	?	?	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	0	0	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
77	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
78	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1
79	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	1	1	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	1
80	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	1	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
81	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
82	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
83	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
84	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0
85	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
86	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	0
87	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1

Table 16: Bit differences that are determined with probability 1 knowing that the difference in round 84 is τ (backward propagation).

rd					δ_4					δ_9		δ_{11}		δ_{13}										δ'_6		δ'_9				k_a	k_b	IR
51	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	0	1
52	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	1
53	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	?	?	0	0	1
54	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	?	0	1	0
55	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0
56	?	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?	1	0	0
57	0	?	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	0	?	?	?	0	?	?	?	?	?	?	?	0	0	0
58	?	0	?	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	1	0	?	?	?	0	?	?	?	?	?	?	0	0	1
59	0	?	0	?	1	?	0	?	?	?	?	?	?	?	?	?	?	?	?	1	0	?	?	?	0	?	?	?	?	?	0	0	0
60	?	0	?	0	?	1	?	0	?	?	?	?	?	?	?	?	?	?	0	?	1	0	?	?	?	0	?	?	?	?	?	0	0	0
61	0	?	0	?	0	?	1	?	0	?	?	?	?	?	?	?	?	?	0	0	?	1	0	?	?	?	0	?	?	?	?	0	0	0
62	0	0	?	0	?	0	?	1	?	0	?	?	?	?	?	?	?	?	0	0	0	?	1	0	?	?	?	0	?	?	?	0	0	0
63	0	0	0	?	0	?	0	?	1	?	0	?	?	?	?	?	?	?	1	0	0	0	?	1	0	?	?	?	0	?	?	0	0	1
64	0	0	0	0	?	0	?	0	?	1	?	0	?	?	?	?	?	?	0	1	0	0	0	?	1	0	?	?	?	0	?	0	0	0
65	0	0	0	0	0	?	0	?	0	?	1	?	0	?	?	?	?	?	?	0	0	1	0	0	0	?	1	0	?	?	?	0	0	1	1
66	1	0	0	0	0	0	?	0	?	0	?	1	?	0	?	?	?	?	?	0	0	0	1	0	0	0	?	1	0	?	?	?	0	0	0
67	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	?	?	?	?	0	0	0	0	1	0	0	0	?	1	0	?	?	0	0	0
68	0	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	?	?	?	0	0	0	0	0	1	0	0	0	?	1	0	?	0	0	0
69	0	0	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	?	?	0	0	0	0	0	0	1	0	0	0	?	1	0	0	1	0
70	0	0	0	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	?	0	0	0	0	0	0	0	1	0	0	0	?	1	0	0	0
71	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	0	0	0	0	0	0	0	1	0	0	0	?	0	0	1
72	0	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	?	1	?	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
73	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	?	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
74	0	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	?	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
75	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
76	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
79	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

D Details on the Incompatibilities for Simon-32/64

Table 17: Bit differences that are determined with probability 1 knowing that the difference in round 9 and 23 and the associated master keys (in blue) are fixed.

Round	Upper State RXD	Upper Key RXD
9	0000 0000 0000 0000 | 0000 0000 0000 0101	0000 0000 0000 0101
10	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
11	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
12	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
13	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
14	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0101
15	0000 0000 0000 0101 | 0000 0000 0000 0000	1111 0000 0000 0101
16	1111 0?0? 0001 ?0?1 | 0000 0000 0000 0101	-
17	0000 0000 0000 0100 | 0000 0?00 0001 ?000	0000 0000 0000 0000
18	0000 0000 0000 0000 | 0000 0000 0000 0100	0000 0000 0000 0100
19	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
20	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
21	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0000
22	0000 0000 0000 0000 | 0000 0000 0000 0000	0000 0000 0000 0001
23	0000 0000 0000 0001 | 0000 0000 0000 0000	-

To check the boomerang constraints from round 16 to 17 we compute:

$$
\begin{aligned}
&(\alpha_\ell \lll 8) \cdot (\delta_r \lll 1) \oplus (\alpha_\ell \lll 1) \cdot (\delta_r \lll 8) \\
&= (11110?0?0001?0?1 \lll 8) \cdot (00000?000001?000 \lll 1) \oplus (11110?0?0001?0?1 \lll 1) \\
&\quad \cdot (00000?000001?000 \lll 8) \\
&= (0000?000001?0000) \oplus (0000?00000000?00) \\
&= 0000?000001?0?00
\end{aligned}
$$

which, no matter the unknown bits, cannot be equal to 0. We thus deduce that there is a contradiction.

E Boomerang Distinguishers Described in [KJK20] on Simon-48/72 and Simon-48/96

Boomerang distinguishers described in [KJK20] on Simon-48/72 and Simon-48/96, together with their claimed probabilities, that have an experimental probability higher than what is claimed.

Table 18: 16-round distinguisher of Simon-48/72 and 18-round distinguisher of Simon-48/96 provided in [KJK20]. (There was a typo in the key at the 7th round for Simon-48/72, 30000c must be changed into 300005 for the following rounds to work as noted).

Round	State RXD	Key RXD	Proba.
5	000005 c50018	f50000	2^{-4}
6	30000c 000005	a00007	2^{-7}
7	000002 30000c	300005	2^{-2}
8	000001 000002	000006	2^{-2}
9	000000 000001	000001	1
10	000000 000000	000000	1
11	000000 000000	000000	1
12	000000 000000	000004	1
13	000004 000000	c00005	2^{-2}
14	c00015 000004	-	
14	000002 30000c	300005	2^{-2}
15	000001 000002	000006	2^{-2}
16	000000 000001	000001	1
17	000000 000000	000000	1
18	000000 000000	000000	1
19	000000 000000	000004	1
20	000004 000000	c00006	2^{-2}
21	c00016 000004	-	

Round	State RXD	Key RXD	Proba.
12	000042 c0011b	c00003	2^{-4}
13	000010 000042	000006	2^{-2}
14	000004 000010	000000	2^{-2}
15	000000 000004	000004	1
16	000000 000000	000000	1
17	000000 000000	000000	1
18	000000 000000	000000	1
19	000000 000000	000001	1
20	000001 000000	300005	2^{-2}
21	300001 000001	-	
21	c00000 900007	100004	2^{-4}
22	800000 c00000	c00003	2^{-2}
23	000001 800000	800004	2^{-2}
24	000000 000001	000001	1
25	000000 000000	000000	1
26	000000 000000	000000	1
27	000000 000000	000000	1
28	000000 000000	000004	1
29	000004 000000	c00005	2^{-2}
30	c00015 000004	-	

Note that the actual probability of the lower characteristic for Simon-48/72 has probability 2^{-6} instead of the 2^{-4} claimed in [KJK20]. This also occurs with the upper characteristic for Simon-32/64, which has probability 2^{-8} instead of the claimed 2^{-6}. This means the naive analysis of the corresponding boomerang distinguishers should have respectively probability 2^{-46} and 2^{-28} instead of the claimed 2^{-42} and 2^{-24}.

F 140-round Distinguisher on KATAN32

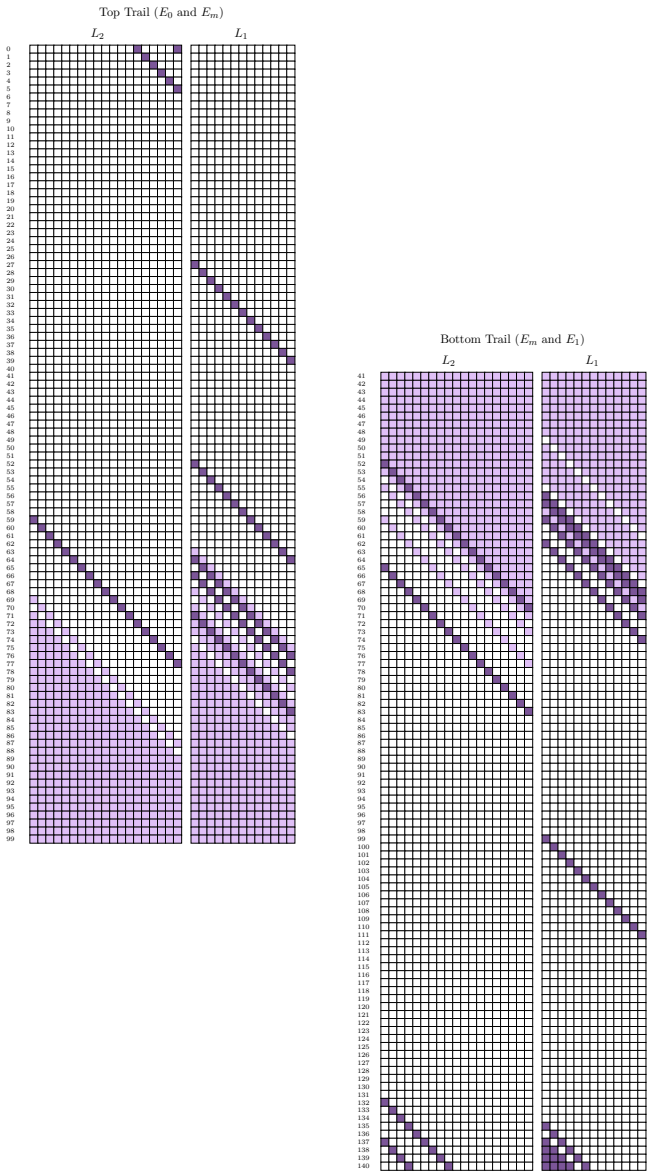

Figure 14: Our 140-round distinguisher on KATAN32 of experimental probability $2^{-16.2}$. White cells are inactive bits, light colored ones are undetermined and darker ones are active. The parameters used are $|E_0| = 40$, $|E_m| = 60$ and $|E_1| = 40$.

G 18-round Distinguisher on Simon-32/64

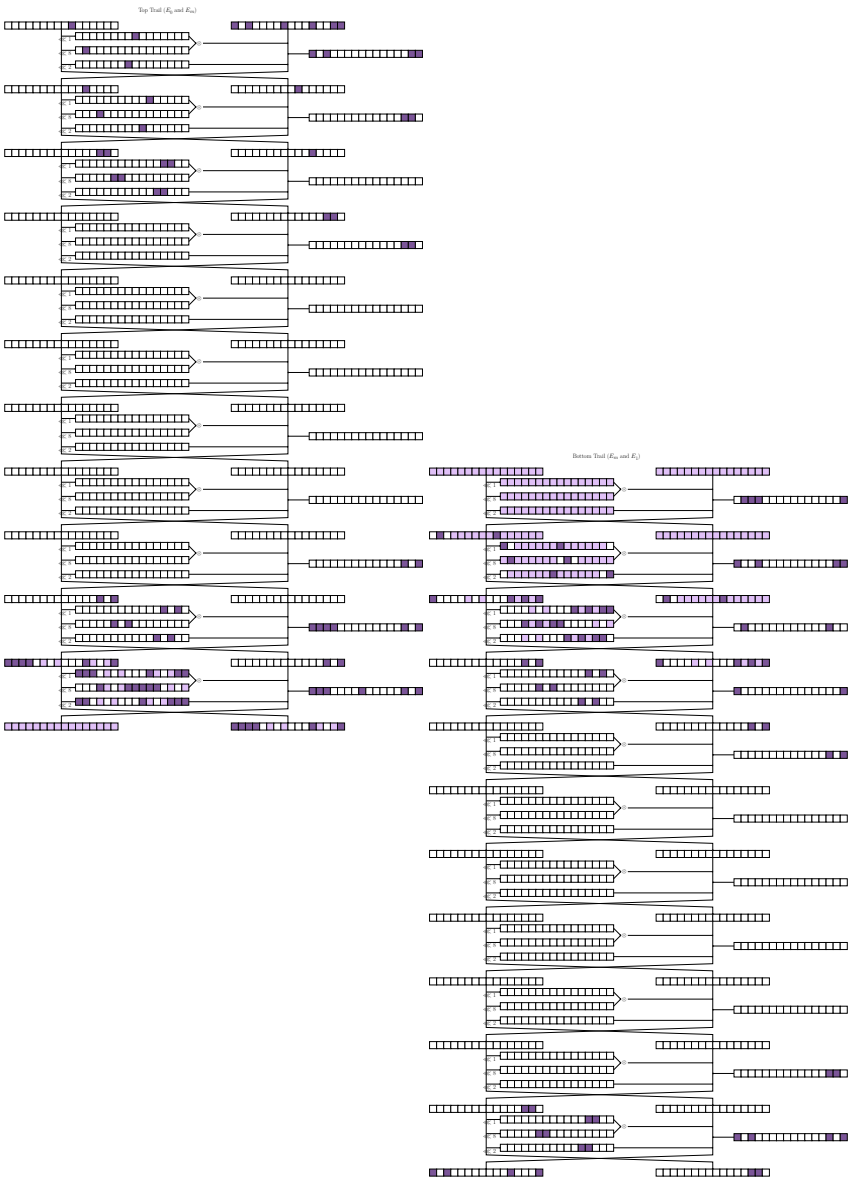

Figure 15: Our 18-round distinguisher on SIMON-32/64 of probability 2^{-24}. White cells are inactive bits, light colored ones are undetermined and darker ones are active. The parameters used are $|E_0| = 6$, $|E_m| = 6$ and $|E_1| = 6$.

H 19-round Distinguisher on Simon-32/64

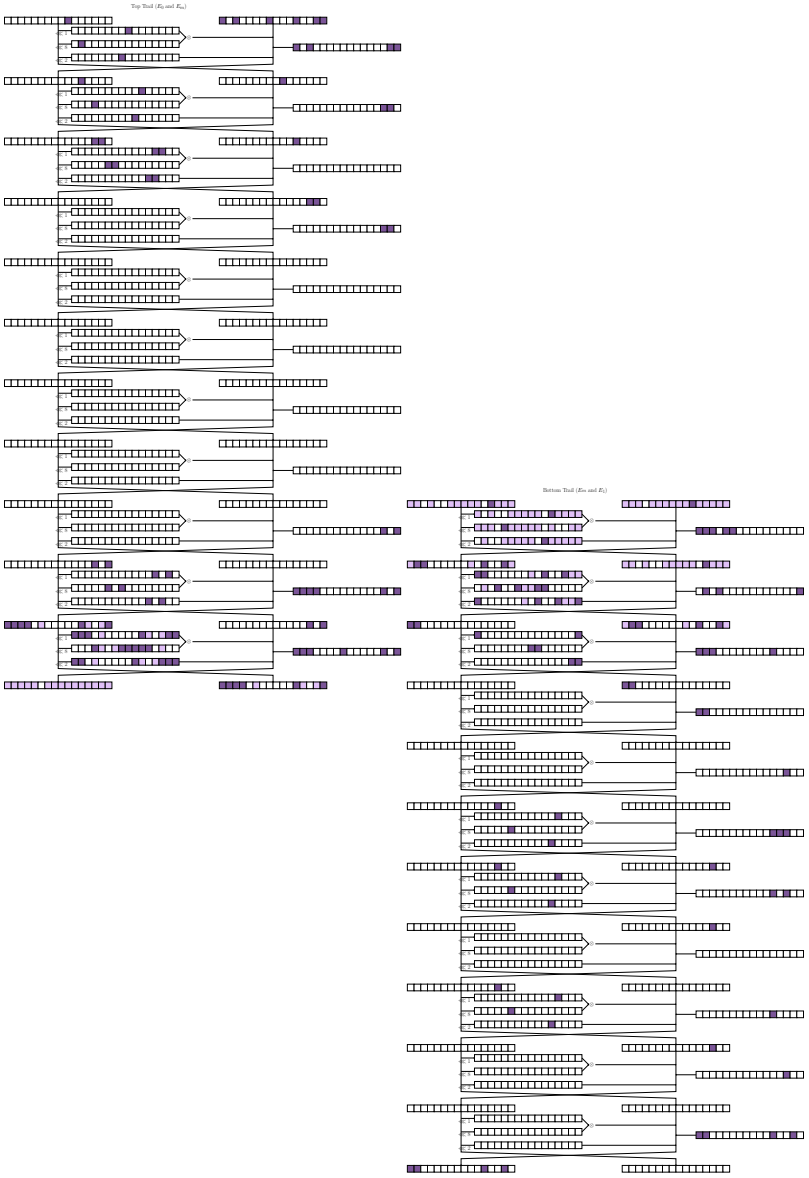

Figure 16: Our 19-round distinguisher on SIMON-32/64 of experimental probability $2^{-29.5}$. White cells are inactive bits, light colored ones are undetermined and darker ones are active. The parameters used are $|E_0| = 7$, $|E_m| = 5$ and $|E_1| = 7$.

I Details on the SMT Model for Simon

This section gives further details on the functioning of our SMT model looking for boomerang distinguishers on SIMON. The model on KATAN has the same structure.

I.1 Parameters

The parameters of the model are:

- the type of attack that is searched. It can be a related-key boomerang, a related-key rotational-xor boomerang or a related-key rotational-xor-differential boomerang,
- $|E_0|, |E_m|$ and $|E_1|$ (the number of rounds of E_0, E_m and E_1) together with the round at which starts the distinguisher,
- the absolute value of the binary logarithm of the probability of the distinguisher.

The SMT solver outputs either `unsat` when no solution with these parameters exists, or `sat` together with the value of all the variables of a valid solution.

I.2 Variables

The 32 internal-state bits are handled in the model with two binary variables each. One that indicates if the difference in this bit is known or not, and the other that stores the value of the difference (when known) so that each bit difference can have a status among $\{0, 1, ?\}$.

The top characteristic variables are represented for the first $|E_0| + |E_m|$ rounds, and the bottom ones over the last $|E_m| + |E_1|$ rounds. Variables are also introduced for the key differences (for the top and bottom characteristics) and for the probability of each round, together with various intermediate variables that will help dealing with the active AND.

I.3 Logical Statements

As we use an SMT solver, we have access to Boolean formulas plus some other data structures and operators, that are grouped in a theory. In particular, we heavily rely on the `bitvector` theory, that gives access to fixed-size Boolean arrays and supports (using the prefix `bv`), besides AND, OR and NOT, the XOR, ADD, MUL, and NEG, which help to implement conveniently integer arithmetic and $\mathbb{F}_2$-linear operations. Utilities also contains operations to rotate an array ($T \ggg i$ is `(_ rotate_right i) T` and $T \lll i$ is `(_ rotate_left i) T`) and extract a sub-array ($T[i..j]$ is `(_ extract j i) T`).

Formulas that must be verified are given to the model with the `assert` command. They tend to be equalities in our case, but this is not mandatory.

Key Schedule. The integration of the equations describing the key derivation are straightforward as the key schedule is linear and as we can directly rely on the bitvector theory.

***E_0* and *E_1*.** We make the assumption that the top and bottom characteristics are independent from each other in the first E_0 rounds and in the last E_1 rounds, so the idea is to look for two fully specified differential characteristics (differences in $\{0, 1\}$ only) over these rounds and to take into account the square of their probabilities in the final cost (these are the usual p^2 and q^2 terms).

Our code for this part is based on the SMT model given in [LLA+20] that looks for rotational-xor characteristics in SIMON-like ciphers.

- The difference of the left branch at round i is copied in the one of the right branch at round $i+1$,
- The close formula of [KLT15] is used to deal with the other branch passing through the round function. As a glimpse of the syntax used in our code, recall that the intermediate variable `varibits` (recalled in Theorem 3) is defined as:

$$\texttt{varibits} = (\alpha \lll 8) \vee (\alpha \lll 1).$$

  To include this equality in our model we simply write:
  `(assert (= varibits (bvor ((_ rotate_left 8) alpha) ((_ rotate_left 1) alpha))))`.
  The variables `doublebits` and γ are similarly easily defined, and only the probability computation remains to be handled. To do so, we start by forcing that $\gamma \cdot \overline{\texttt{varibits}} = 0$ with: `(assert (= #x0000 (bvand (bvxor varibits #xffff) gamma)))`. The condition $\gamma \oplus \gamma \lll (8-1)) \cdot \texttt{doublebits} = 0$ is given by:
  `(assert (= #x0000 (bvand (bvxor ((_ rotate_left 7) gamma) gamma) doublebits)))`
  and finally we compute `(assert (= z (bvxor varibits doublebits)))`. The probability of the difference transition is then obtained from the weight of z.
- The round ends with the computation of the left branch.

E_m. The $|E_m|$ middle rounds contain all the dependencies between the top and bottom characteristics. Contrary to what we set for E_0 and E_1 and in order to take into account several valid characteristics, the differences in E_m lie in the set $\{0, 1, ?\}$. The two things ensured in these rounds are:

- *Correct propagation of (partial) differences:* the difference characteristic of E_0 is extended forward over the E_m middle rounds, and the difference characteristic of E_1 is extended backward over the E_m middle rounds. We tweak the algorithm given in [KLT15] to allow that some difference bits are not fixed.
- *Compliance with the boomerang constraints:* We enforce that the boomerang constraint is fulfilled in every round, that is, that the equality given in Equation (2), Equation (3) or Equation (4) (depending on the type of attack that is considered) is verified.

More into details, our implementation of the (partial) difference propagation in E_m works as follows (at the bit level):

1. If the AND is inactive:
   (a) The probability cost of the AND is 2^0.
   Check if the linear bits used to compute the left difference one round later are all determined (not '?'):
   - If they are, set the left difference one round later to the xor of these bits
   - Else set the left difference one round later to '?'
2. Else:
   (a) The output bit difference is fixed, the probability cost is 2^{-1}.
   - If the fixed AND output is correlated to a second AND with a fixed output too and if the common bit is inactive, the overall cost is increased by a factor of 2^1 (this corresponds to the `doublebits` of [KLT15]) and the two output differences are equal.

   (b) Or the output bit difference is left unknown ('?') and in this case the probability cost is 2^0.

Probability. Finally, the last equality that is enforced is between the value of the probability given in parameter by the user and the one computed for the E_0, E_1 and E_m parts.

I.4 Coding Techniques

We detail here some tricks we used to implement the model.

Conditional assertions. We have many cases where we want to express "if X, then Y = Z". We model it as "X AND Y = X AND Z", that is, `assert( = (∧ X Y) (∧ X Z))`. For example, this allows to express equations that hold only if the bit differences are known.

Packing rotation-invariant assertions. Many bit constraints in the model for SIMON are rotation-invariant, and we have to model many constraints that are identical up to a shift of the bit indexes. We pack all these bit constraints into one vectorial constraint: we take one constraint that involves bit x_0 for an arbitrary vector x, and we replace all the bits y_i involved in the constraint by a rotation of the vector y by i bits. For example, if we have $x_i \vee y_{i+3 \mod n} = y_{i+1 \mod n} \wedge z_{i+2 \mod n}$, for all i, the equations can be packed as $x \vee (y \lll 3) = (y \lll 1) \wedge (z \lll 2)$.

Note that the same can be done for shifts of indexes: the equations $x_i \vee y_{i+3} = y_{i+1} \wedge z_{i+2}$ for $0 \leq i \leq n-4$ can be packed as $x[0..n-4] \vee y[3..n-1] = y[1..n-3] \wedge z[2..n-2]$. This approach is used to implement KATAN's key schedule.