Differential Analysis and Meet-in-the-Middle Attack against Round-Reduced TWINE

Alex Biryukov, Patrick Derbez, and Léo Perrin[⋆]
{alex.biryukov,patrick.derbez,leo.perrin}@uni.lu

SnT, University of Luxembourg

Abstract. TWINE is a recent lightweight block cipher based on a Feistel structure. We first present two new attacks on TWINE-128 reduced to 25 rounds that have a slightly higher overall complexity than the 25-round attack presented by Wang and Wu at ACISP 2014, but a lower data complexity.
Then, we introduce alternative representations of both the round function of this block cipher and of a sequence of 4 rounds. LBlock, another lightweight block cipher, turns out to exhibit the same behaviour. Then, we illustrate how this alternative representation can shed new light on the security of TWINE by deriving high probability iterated truncated differential trails covering 4 rounds with probability 2^{-16}.
The importance of these is shown by combining different truncated differential trails to attack 23-rounds TWINE-128 and by giving a tighter lower bound on the high probability of some differentials by clustering differential characteristics following one of these truncated trails. A comparison between these high probability differentials and those recently found in a variant of LBlock by Leurent highlights the importance of considering the whole distribution of the coefficients in the difference distribution table of a S-Box and not only their maximum value.
Keywords: TWINE, LBlock, meet-in-the-middle, truncated differential, cryptanalysis

1 Introduction

Lightweightness is currently one of the most investigated topics in symmetric cryptography. As more and more appliances are expected to communicate with each other as well as over the internet, the need for primitives capable of running on low-power CPU's e.g. used in sensor networks as well as on small RFID tags is becoming more pressing. Many lightweight primitives intended to be usable in such constrained environment have been proposed during the last few years, a review of which can be found in [1].

A possible approach to design a lightweight primitive is to use many rounds with a simple structure. The Generalized Feistel Network (GFN), introduced by Nyberg in [2], is a modification of the regular Feistel Network which uses

[⋆] Patrick Derbez and Léo Perrin are supported by the CORE ACRYPT project from the *Fond National de Recherche* (Luxembourg).

more than 2 branches. Having more branches allows the use of a simpler Feistel function, the branch permutation taking care of the diffusion, hence the suitability of this approach in a constrained context. However, the simple branch rotation used in most GFN with b branches requires b rounds to obtain full diffusion. To improve this number, more sophisticated permutations were introduced in [3] and one such permutation has been used by the authors of TWINE [4], a lightweight block cipher with a GFN structure: while TWINE uses 16 branches, only 8 rounds are necessary for full diffusion. TWINE is therefore both a good example of common trade-offs in lightweight cryptography, e.g. it has a simple round function iterated many times, and one of the only instances of a GFN with improved diffusion layer. A similar block cipher is LBlock [5], a lightweight block cipher which served as the basis for the design of LBlock-s, a variant with a different S-Box and key schedule used in the Lightweight Authenticated Cipher (LAC) submitted to the CAESAR competition by a related team [6]. While LBlock is described as a "regular" two-branched Feistel Network, the rotation used in its permutation layer and the simplicity of its Feistel function make it equivalent to a GFN similar to TWINE. The designers of TWINE pointed out this resemblance in [4].

In this paper, we focused our efforts on TWINE and tried different approaches to cryptanalyze it. First, we study Meet-in-the-Middle (MitM) attacks on TWINE-128 and describe an attack on 25 rounds[1]. It is based on the attack strategy proposed by Demirci and Selçuk at FSE 2008 [9] to attack both the 192 and 256-bit version of the AES reduced to 8 rounds and which is the starting point of the best attacks on the AES so far [10,11,12]. Then we study impossible differential attacks and show that thanks to the framework described by Boura *et al.* in [13] one can be mounted on 25 rounds with an overall complexity below the natural bound of the exhaustive search. Our 25-round attacks have a slightly higher time complexity than the 25-round attack presented by Wang and Wu [14] at ACISP 2014 but a lower data complexity. Interestingly, three different cryptanalysis techniques (meet-in-the-middle, impossible differential and zero-correlation linear) allow to break the same number of rounds with a similar overall complexity.

The particular permutation layer of TWINE implies, as we will see, an observable vulnerability of this block cipher against truncated differential cryptanalysis, an attack introduced by Knudsen [15]. Unlike "normal" differential cryptanalysis, this technique does not rely on studying fully specified trails where each bit of difference is supposed to have a particular value but instead on looking at more general patterns where some bit differences may take both values 0 and 1. In the case of word oriented cipher, we can restrict the investigation to trails where the differences are studied at the word level: either there is at least one difference over the whole word or there is none. Trails where some of the bits are not specified are often used when adding rounds on top and on the bottom

[1] While a MitM attack on 25-round TWINE-128 is already in the literature [7], it has been shown in a note on eprint [8] that the complexity of this attack is actually higher than brute-force.

of a differential distinguisher. However, using truncated differential covering all the rounds can also yield powerful attacks. For example, such an approach has been used recently by Lallemand et al. [16] to attack the lightweight block cipher KLEIN [17]. Truncated differential have also been used to enhance the search for high probability differentials. Two recent examples are the best attack on the block cipher PRINCE [18] and a differential forgery attack on the authenticated cipher LAC [19].

As we introduce new attacks on TWINE, we summarize the complexities of the best attacks against this cipher in the single-key model in Table 1.

Description			Complexity		
Reference	Type	Version	Data	Time	Memory
[20]	Biclique	full TWINE-80	2^{60}	$2^{79.1}$	2^{8}
		full TWINE-128	2^{60}	$2^{126.82}$	2^{8}
[21]	Impossible diff.	23r TWINE-80	$2^{57.85}$	$2^{79.09}$	$2^{78.04}$
		24r TWINE-128	$2^{58.1}$	$2^{126.78}$	$2^{125.61}$
[14]	Zero-Cor. Linear	23r TWINE-80	$2^{62.1}$	$2^{72.15}$	2^{60}
		25r TWINE-128	$2^{62.1}$	$2^{122.12}$	2^{60}
Section 3.1	MitM	25r TWINE-128	2^{48}	$2^{124.7}$	2^{109}
Section 3.2	Impossible diff.	25r TWINE-128	$2^{59.1}$	$2^{124.5}$	$2^{78.1}$
Section 5.3	Truncated diff.	23r TWINE-128	2^{58}	$2^{126.78}$	2^{89}
			2^{62}	$2^{125.94}$	
			2^{64}	$2^{124.35}$	

Table 1: The best attacks on TWINE in the single-key model.

Our Contributions First, we describe in Section 3 our best attacks on TWINE-128, namely both a Meet-in-the-Middle attack and an Impossible Differential attack, leveraging the simplicity of the key schedule of this block cipher.

Then, we highlight in Section 4 a property of the permutation used in TWINE: rounds of encryption can be grouped into blocks of 4 rounds in such a way that two halves of the internal states of both ciphers evolve independently from one another during the first 3 rounds of the block and exchange information only during the fourth. We also discuss why LBlock and its simpler variant LBlock-s exhibit the same 4-round behaviour. As a consequence of this observation, we describe several high probability truncated differential trails for all these ciphers. We then leverage them in Section 5 to attack 23 rounds of TWINE-128 using comparatively low memory. Finally, we use these truncated trails to optimize a search for high probability differentials and show that the conservative choice of S-Box made by the designers of TWINE greatly limits the differential effect in this primitive — unlike in LBlock-s for instance.

2 Descriptions of TWINE, LBlock and LBlock-s

2.1 Description of TWINE

This block cipher uses 16 branches of 4-bits and has a very simple round function (see Figure 1): the Feistel function consists in a xor of a sub-key and a call to a unique S-box based on the inverse function in $GF(2^4)$. Then, the branches are shuffled using a sophisticated nibble permutation ensuring faster diffusion than a simple shift [3]. One version of TWINE uses an 80 bits key, another uses a 128 bits key and we denote these versions TWINE-80 and TWINE-128. They only differ by their key-schedule and both have 36 rounds. Both key schedules are sparse GFN's using only 2 S-Box calls per round for TWINE-80 and 3 for TWINE-128. At each round, some fixed nibbles of the key-state are used as round keys for the block cipher. One round of TWINE is depicted on Figure 1.

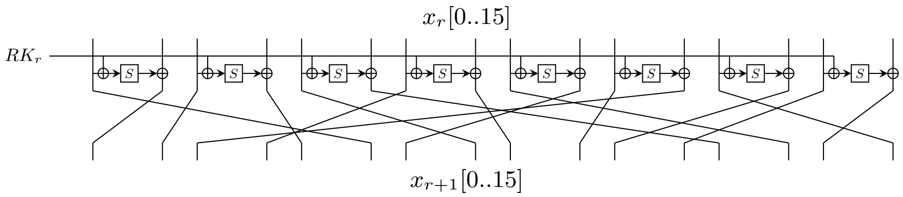


Fig. 1: The round function of TWINE.

Notations. Given a collection of messages $\{P^0, \ldots\}$, the nibble with index i taken at round r of message m is denoted $x_r^m[i]$. The master key is denoted K while the round key used at round r is denoted RK_r.

Keyschedule. The keyschedule produces the 36 round keys from the master key K. It is a variant of GFN with few Sboxes which is the same as the one used in the round function of TWINE. Two key lengths are available: 80 and 128 bits. In both cases, the subkey WK_0 is first initialized to K and then next subkeys are generated using round constants and the same round function: $WK_{i+1} = F(WK_i, CON^i)$, for $0 \leq i \leq 31$. Finally the round key RK_i is obtained by extracting 8 nibbles from WK_i. The function F used for 128-bit keys is depicted on Figure 2. We refer the reader to [4] for the 80-bit version of the keyschedule.

2.2 Descriptions of LBlock and LBlock-s

LBlock [5] is a two-branched Feistel Network with a twist: a rotation is performed on the branch being xor-ed with the output of the Feistel function. This leads to a strong structural proximity with TWINE, as the authors of this cipher acknowledged.

The Feistel function of LBlock is made of a key addition, a S-box layer S made of 8 different 4-bits S-boxes and a nibble permutation P. In addition to the

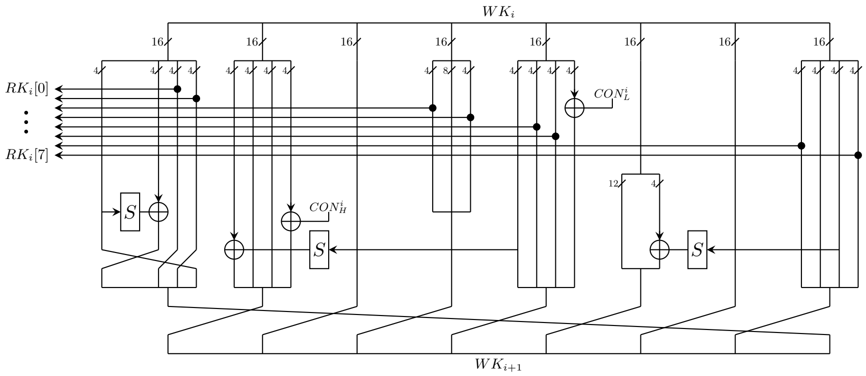


Fig. 2: Keyschedule of TWINE-128.

usual Feistel structure, there is a rotation by 8 bits to the left on the right branch before the xor. The complete round function is described in Figure 3. LBlock only uses 80-bits keys. Its key-schedule is similar to that of PRESENT [22]: it relies on a rotation of the 80-bits register used to store the master key and on the application of two S-boxes. It uses 32 rounds to encrypt a plaintext.

LBlock-s, the block cipher used in the authenticated cipher LAC [6], is identical to LBlock except that the S-Box layer uses a unique S-Box instead of 8 different ones and that its key-schedule is closer to the one of TWINE-80. The S-Boxes of LBlock and that of LBlock-s all have similar differential properties.

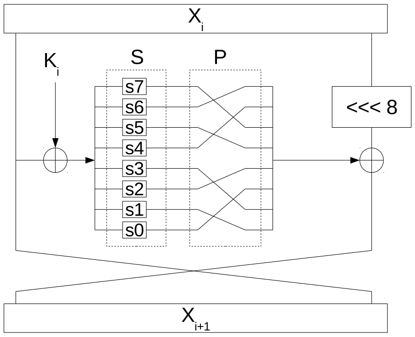


Fig. 3: The round function of LBlock

3 New Attacks on 25-Round TWINE-128

In this section we present two new attacks on 25-round TWINE, increasing by one the number of rounds broken if we omit biclique attacks.

3.1 Meet-in-the-Middle Attack on 25-Round TWINE-128

Our meet-in-the-middle attack follows the strategy used by Demirci and Selçuk on AES in [9], later improved by Dunkelman et al. in [23], Derbez et al. in[11,10] and by Li et al. in [12]. That is the first time that this kind of meet-in-the-middle attack is applied to a Feistel Network and this shows that this technique is also powerful on such ciphers.

First we give the definition of a δ-set which is a particular structure of messages used in our attack.

Definition 1. *Let a δ-set be a set of 16 TWINE-states that are all different in one state nibble (the active nibble) and all equal in the other state nibbles (the inactive nibbles).*

In the following we consider δ-sets such that the nibble 15 is the active one. For such a particular set we made the following observation which is the core of our new attack.

Observation 1 *Consider the encryption of a δ-set $\{P^0, P^1, \ldots, P^{15}\}$ through eleven full TWINE rounds. The ordered sequence*

$$[x_{11}^1[4] \oplus x_{11}^0[4], x_{11}^2[4] \oplus x_{11}^0[4], \ldots, x_{11}^{15}[4] \oplus x_{11}^0[4],$$
$$x_{11}^1[15] \oplus x_{11}^0[15], \ldots, x_{11}^{15}[15] \oplus x_{11}^0[15]]$$

is fully determined by the following 27 nibble parameters:

- $y_1^0[14]$
- $y_2^0[14]$
- $y_3^0[2, 14]$
- $y_4^0[2, 4, 14]$
- $y_5^0[0, 2, 4, 14]$
- $y_6^0[0, 2, 8, 12, 14]$
- $y_7^0[0, 4, 6, 10, 14]$
- $y_8^0[2, 8, 12]$
- $y_9^0[4, 10]$
- $y_{10}^0[2]$

where $y_r^m[2i] = x_r^m[2i] \oplus RK_r[i]$. Consequently, there are at most $2^{4\times 27} = 2^{108}$ possible sequences when we consider all the possible choices of keys and δ-sets (out of the $2^{4\times 2\times 15} = 2^{120}$ of the theoretically possible 30-nibble sequences).

Proof. The proof is straightforward and depicted on Figure 4. At the first step we know the differences $P^1 \oplus P^0, \ldots, P^{15} \oplus P^0$. As we are considering a δ-set, the differences in each sbox of the first round are null and thus we are able to compute the differences $x_1^1 \oplus x_1^0, \ldots, x_1^{15} \oplus x_1^0$. So the knowledge of $y_1^0[14]$ leads to

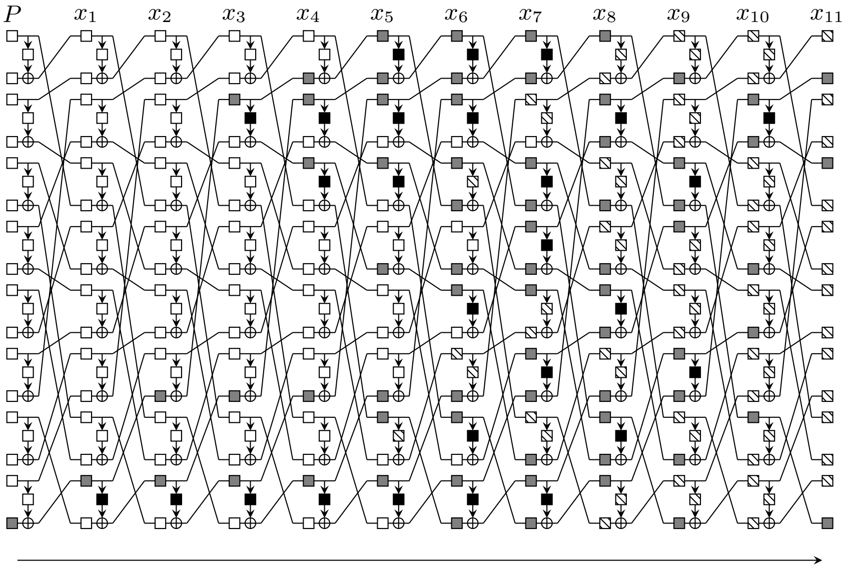


Fig. 4: Encryption of a δ-set through 11 full TWINE rounds. Black nibbles are the parameters given in Observation 1. Differences in coloured nibbles are known. No difference in white nibbles.

the knowledge of this particular state variable for all the 16 messages and thus we know the differences in each sbox of this round and are able to compute the differences $x_2^1 \oplus x_2^0, \dots, x_2^{15} \oplus x_2^0$. This procedure can be repeated until differences in both $x_{11}[4]$ and $x_{11}[15]$ are reached since at each step differences in sboxes are either null, not required or known.

Note that the actual value of the active nibble of P^0 does not affect the set of all the possible sequences since only differences are used. Thus the choice of P^0 is free but then the δ-set has to be ordered according to the difference in the active nibble.

This observation on 11-round TWINE is used to mount an attack on 25-round TWINE by adding 5 rounds at beginning and 9 at the end. The scenario of the attack is the following:

- **Offline phase.** Compute all the 2^{108} 120-bit sequences given in Observation 1, and store them in a hash table.
- **Online phase.**
  1. Pick a plaintext P^0.
  2. Guess the state variables required to identify a δ-set containing P^0.
  3. Ask for the corresponding ciphertexts.
  4. Guess the state variables required to compute differences in both $x_{11}[4]$ and $x_{11}[15]$ from the ciphertexts.
  5. Build the sequence and check if it belongs to the table.

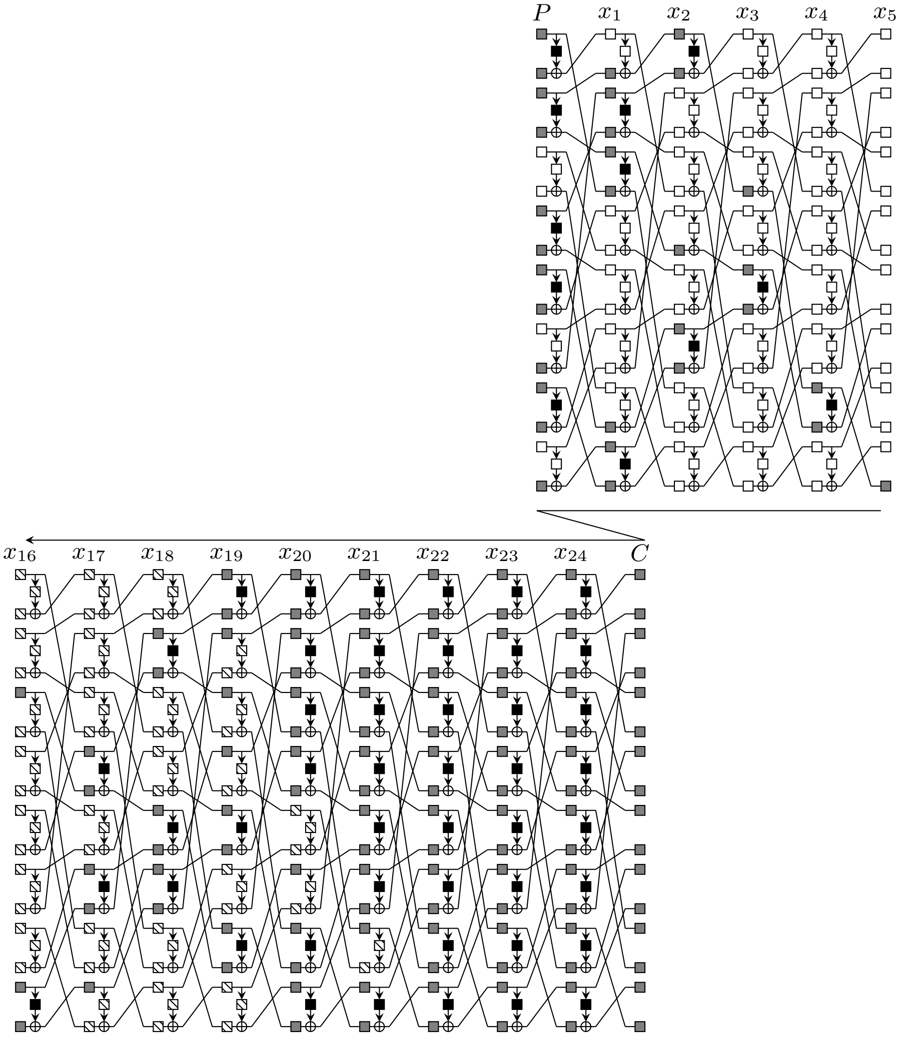


Fig. 5: Online phase of the 25-round attack. Black nibbles have to be known to compute differences in all coloured nibbles. No difference in white nibbles.

Steps 2 and 4 are similar to the proof of Observation 1: first we propagate the differences from state x_5 to the plaintext and then we propagate differences from the ciphertexts to both $x_{11}[4]$ and $x_{11}[15]$. Thus 58 state nibbles are needed to perform the online phase as depicted on Figure 5. Hopefully, the keyschedule equations reduce the amount of possible values from $2^{4 \cdot 58} = 2^{232}$ to 2^{124}. Indeed, knowing the full subkey WK_6 except nibble 26 leads to the knowledge of enough key material to partially encrypt and decrypt the plaintext and the ciphertext in order to obtain the value of the required state variables. This key material is depicted on Figure 6.

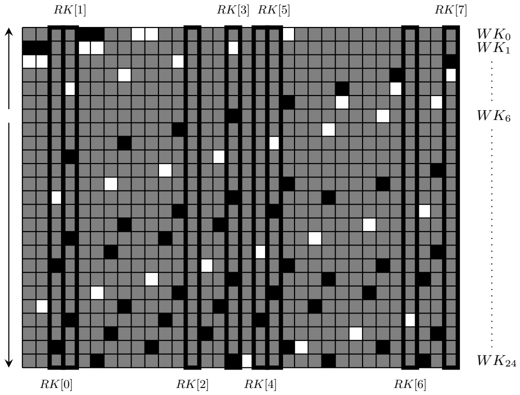


Fig. 6: Subkeys of 25-round TWINE. Gray (resp. colored) nibbles are computed from the full WK_6 except nibbles 15 and 26 (resp. except nibble 26).

The data complexity of this attack is 2^{48} chosen plaintexts, the time complexity is $2^{124} \cdot 16$ partial encryptions/decryptions and the memory complexity is around 2^{108} 128-bit sequences. The probability for a false positive is approximately $2^{108} \cdot 2^{-120} = 2^{-12}$ and, as we try 2^{124} key guess, we expect that only 2^{116} remain after the last step. Thus, one can guess $WK_6[26]$ to fully recover the master key and then test it against two plaintext/ciphertext pairs.

Note that some minor improvements can be applied to the attack. First we can consider δ-set of 15 messages instead of 16 to save some memory and time complexity while still providing enough filtering to retrieve the master key without increasing the overall complexity. Furthermore, knowing the subkey WK_6 except nibbles 15 and 26 provides enough key material (gray colored on Figure 6) to compute all the state variables required by step 2 together with all the ones required by step 4 except 21 of them. Those ones are $y_{16}[14]$, $y_{17}[6, 10]$, $y_{18}[2, 8, 10]$, $y_{19}[0, 12]$, $y_{20}[2, 4, 6, 14]$, $y_{21}[2, 6, 8]$, $y_{22}[0, 2]$, $y_{23}[0, 4]$ and $y_{24}[6]$. Hence, we estimate the time complexity to be:

$$2^{120} \cdot 15 \cdot 37/200 + 2^{124} \cdot 15 \cdot 21/200 + 2 \cdot 2^{120} \approx 2^{124.7} \text{ encryptions},$$

where 200 is the number of sboxes for one encryption. The memory complexity is approximately 2^{109} 64-bit blocks.

3.2 Impossible Differential Attack on 25-Round TWINE-128

Impossible differential cryptanalysis simultaneously introduced by Knudsen [24] and Biham *et al.* [25] is a powerful technique against a large variety of block ciphers. Recently, Boura *et al.* [13] proposed a generic vision of impossible differential attacks with the aim of simplifying and helping the construction and verification of this type of cryptanalysis. In particular, they provided a formula to compute the complexity of such an attack according to its parameters. To understand the formula we first briefly remain how an impossible differential attack is constructed. It starts by splitting the cipher in three parts: $E = E_3 \circ E_2 \circ E_1$ and by finding an impossible differential ($\varDelta_X \nrightarrow \varDelta_Y$) through E_2. Then $\varDelta_X$ (resp. $\varDelta_Y$) is propagated through E_1^{-1} (resp. E_3) with probability 1 to obtain $\varDelta_{in}$ (resp. $\varDelta_{out}$). We denote by c_{in} and c_{out} the $\log_2$ of the probability of the transitions $\varDelta_{in} \to \varDelta_X$ and $\varDelta_{out} \to \varDelta_Y$ respectively. Finally we denote by k_{in} and k_{out} the key materials involved in those transitions. All in all the attack consists in discarding the keys k for which at least one pair follows the characteristic through E_1 and E_3 and in exhausting the remaining ones. The complexity of doing so is the following:

- **data:** C_{N_α}
- **memory:** N_α
- **time:** $C_{N_\alpha} + \left(1 + 2^{|k_{in} \cup k_{out}| - c_{in} - c_{out}}\right) N_\alpha C_{E'} + 2^{|k| - \alpha}$

where N_α is such that $(1 - 2^{-c_{in} - c_{out}})^{N_\alpha} < 2^{-\alpha}$, C_{N_α} is the number of chosen plaintexts required to generate N_α pairs satisfying $(\varDelta_{in}, \varDelta_{out})$, $|k|$ is the key size and $C_{E'}$ is the ratio of the cost of partial encryption to the full encryption.

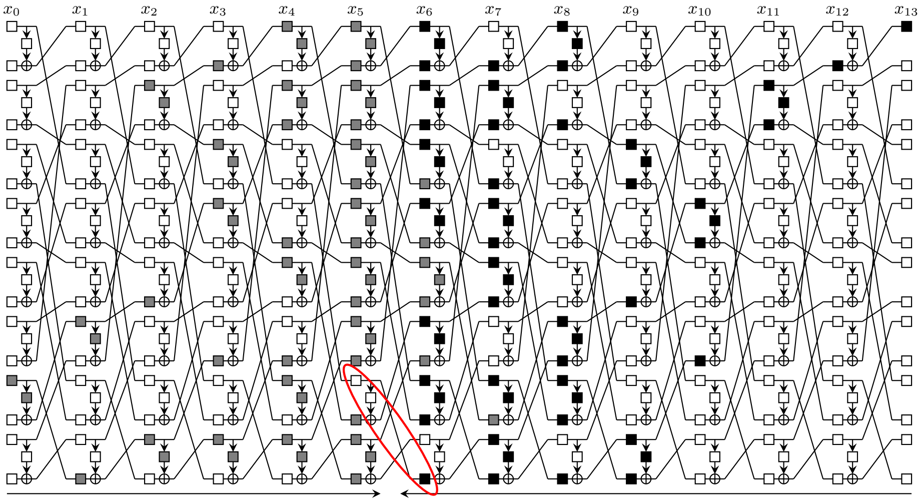


Fig. 7: Impossible truncated differential on 13 TWINE-rounds. No difference in white nibbles. Differences in black (resp. gray) nibbles are (resp. may be) non-zero.

We used this framework to mount an impossible differential attack on 25-round TWINE-128. First we found a truncated impossible characteristic through

13 rounds of TWINE which is described on Figure 7. It was extended by 4 rounds at the start and by 8 rounds at the end in order to attack 25 rounds of the cipher. It can be seen in Figure 8 that the difference in the plaintexts has to be zero in 11 nibbles such that $c_{in} + c_{out} = 16 + 60 = 76$. The key material $k_{in} \cup k_{out}$ is composed of $7 + 45 = 52$ round-key nibbles which can assume only 2^{124} thanks to the keyschedule of TWINE-128. Indeed, they all can be computed from the whole subkey WK_{24} except nibble 1 (see Figure 9).

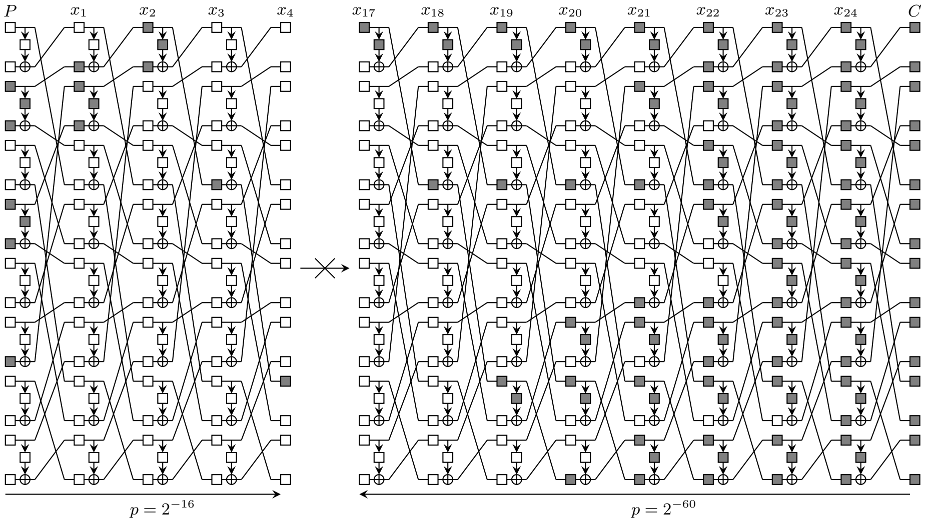


Fig. 8: Impossible differential attack on 25 rounds. No difference in white nibbles.

As a consequence, and according to the above formula, the complexity of our attack is $D = \alpha \cdot 2^{75.5-39} \cdot 2^{20} = \alpha \cdot 2^{56.5}$, $M = \alpha \cdot 2^{75.5}$ and $T \approx \alpha \cdot 2^{123.5} \cdot C_{E'} + 2^{128-\alpha}$. As we estimate the ratio $C_{E'}$ to $52/200 \approx 2^{-1.9}$, the value of α minimizing the overall complexity is 5.87.

4 The 4-Round Structure of TWINE, LBlock and LBlock-s

4.1 Alternative Representation of the Round Functions

The round functions of TWINE can be described using an equivalent representation which allows a clearer representation of some differential paths. This alternative representation is given in Figure 10a. Note that a similar representation of LBlock can be obtained, an observation which highlights the similarities between these two designs.

For TWINE, we simply move all the branches "going" in the Feistel functions to the left and those receiving its output to the right. This means we simply

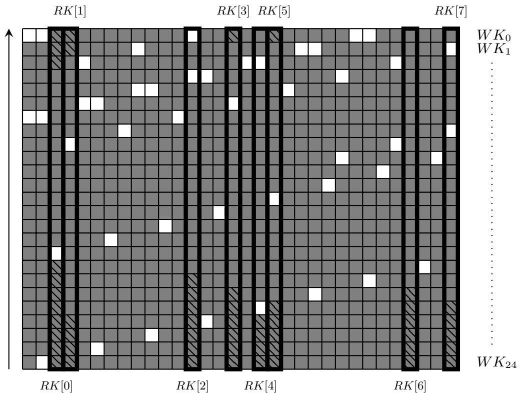
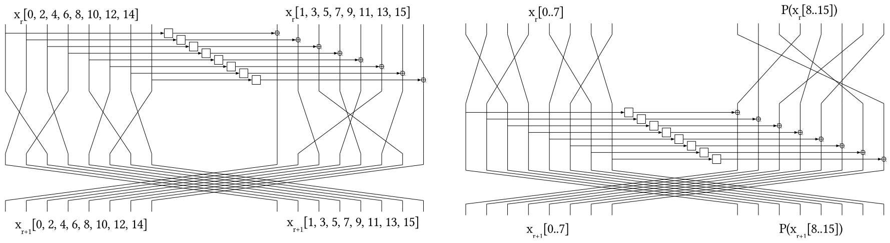


Fig. 9: Subkey nibbles obtained from WK_{24} except nibble 1. Hatched nibbles are the ones required in the impossible differential attack.


x_r[0, 2, 4, 6, 8, 10, 12, 14]
x_r[1, 3, 5, 7, 9, 11, 13, 15]
x_{r+1}[0, 2, 4, 6, 8, 10, 12, 14]
x_{r+1}[1, 3, 5, 7, 9, 11, 13, 15]
x_r[0..7]
P(x_r[8..15])
x_{r+1}[0..7]
P(x_{r+1}[8..15])

(a) TWINE

(b) LBlock(-s)

Fig. 10: Alternative representations of the round functions of TWINE and LBlock(-s).

move branches with even indices on the left and those with odd ones on the right, as described in Figure 10a.

The process leading to the alternative description of LBlock(-s) is more complicated than for TWINE and is summarized in Figure 11. Since the S-boxes and the permutation layer P both operate on nibbles, $P \circ S$ is equivalent to $S' \circ P$ where S' is a reordered S-box layer. Then, instead of applying P within the Feistel function, we apply it before entering it and then apply the inverse $1/P$ of P on the same branch to compensate. Finally, we note that the rotation R and the inverse permutation $1/P$ are applied on the same data, so we combine them into one operation $R \circ (1/P)$. If we replace the two 32-bit words making the internal state of LBlock by eight 4-bits nibbles each, we obtain the representation given in Figure 10b.

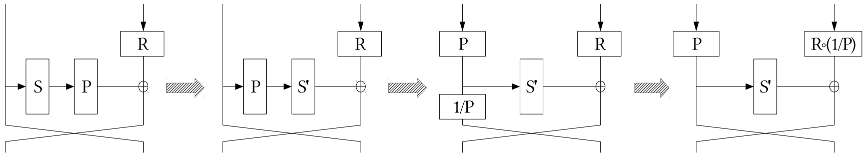


Fig. 11: How to obtain the alternative representation of LBlock(-s).

4.2 A 4-Round Cyclic Behavior

Using our alternative representation, we represent 4 rounds of TWINE easily (see left of Figure 12). As we can see, the 16 branches can be grouped in two disjoint *components*, gray and black, such that branches from one component interact only with each other during 3 rounds out of 4. However, during the last round, branches from each component interact only with branches from the other component. Furthermore, these components are stable in the sense that such groups of 4 iterations can be plugged together to cover any number of rounds and remain separated for all rounds with index r with $r \not\equiv 3 \mod 4$. Indeed, in Figure 12, the branches which are black at the output of the fourth round are exactly those which are gray at the input of the first round. If we draw these components separated from one another, we obtain another description of 4 rounds of TWINE given on the right of Figure 12. The same can be done with LBlock(-s), see Figure 13.

5 Truncated Differential Cryptanalysis of TWINE

5.1 Truncated Differentials over 4 Rounds

Because of the particular structure it has over 4 rounds, TWINE exhibits some truncated differential patterns with high probability. The simplest one implies 4 active branches in input and 4 active branches in the output of 4 rounds at the cost of 4 difference cancellations at round 3. Let $(x[0], x[2], x[6], x[10])$ have non zero differences. Then these differences will propagate to the full black component during the next two rounds. During round 3, if the differences in $(x[0], x[4], x[6], x[12])$ cancel themselves with the differences in $(x[1], x[5], x[7], x[13])$ after going through the key addition and the S-box layer, then the differences do not propagate to the red component. Hence, the differences remain contained in the black component for another 3 round with probability 1. Since 4 cancellations happen with probability 2^{-16} and since such truncated characteristics can be "plugged" so as to cover as many rounds as we want, we have a truncated differential covering $4r$ rounds with probability $2^{-16 \cdot r}$.

Other slightly different characteristics involve three active branches in the input and the output after 4 rounds in such a way that only 4 cancellations are necessary, meaning that they also have a probability of 2^{-16}. One of them is described in Figure 14a and the others in the Appendix in Figure 17. Non-zero

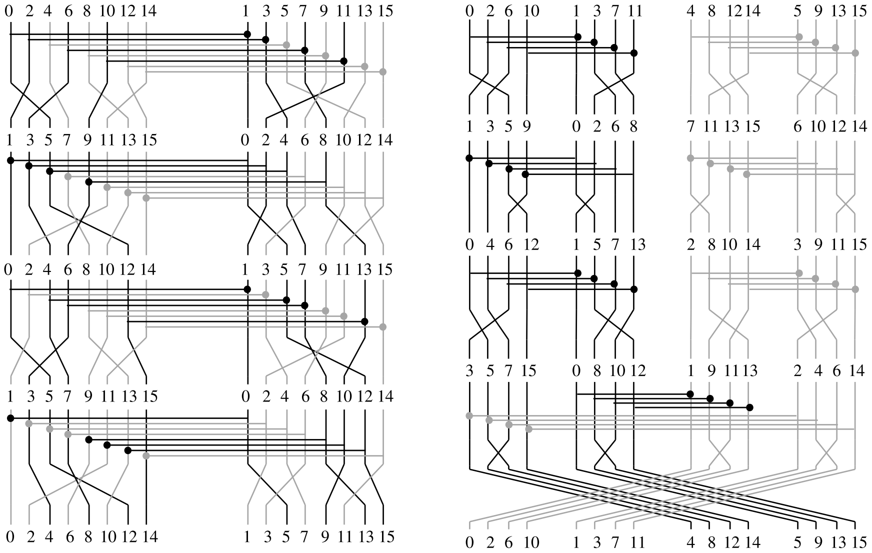


Fig. 12: Alternative representation of 4 rounds of TWINE. S-boxes are not shown and XOR's are represented by circles. On the left is the basic representation, on the right one which highlights the two components. Numbers correspond to nibble indices in the "regular" representation.

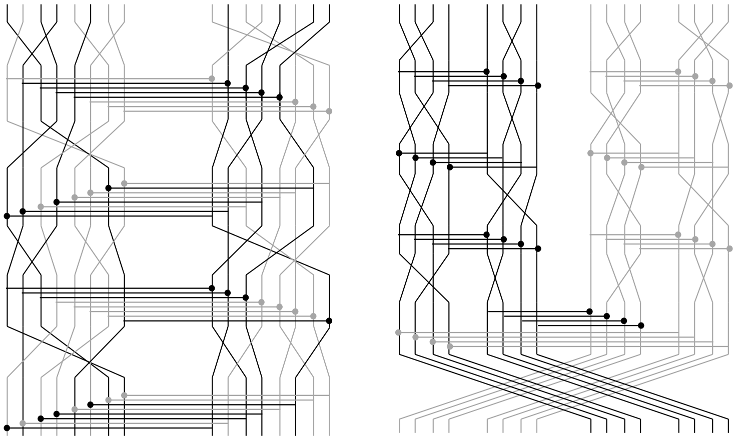
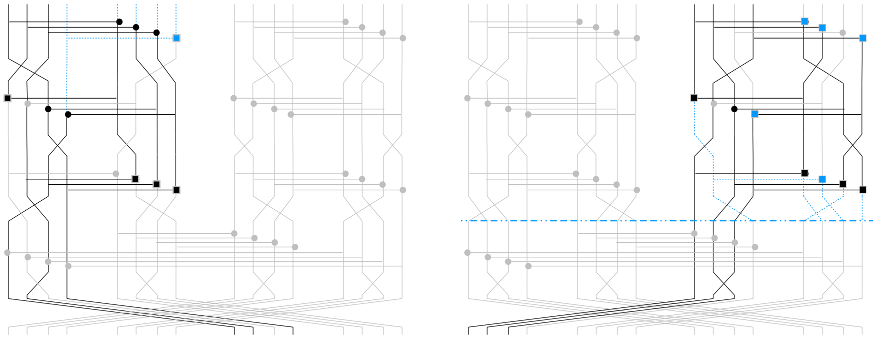


Fig. 13: Alternative representation of 4 rounds of LBlock(-s). S-boxes are not shown and XOR's are represented by circles. On the left is the basic representation, on the right one which highlights the two components.



(a) $\mathcal{L}_1$ (black) and its counterpart $\mathcal{L}'_1$ (b) $\mathcal{R}_1$ (black) and its counterpart $\mathcal{R}'_1$

Fig. 14: 4-round truncated differentials for TWINE and their modified versions.

differences are black and zero differences are gray. They all work by having one cancellation during the second round and three during the third. As before, the first and fourth rounds have probability 1. However, we can extend them for the first 4 rounds by adding non-zero differences over all the components (which is represented in a light blue dotted line in Figure 17). At the cost of one more cancellation, hence a probability of 2^{-20}, we can use structures made of 2^{32} plaintext/ciphertext couples giving raise to $\binom{2^{32}}{2} \approx 2^{63}$ pairs with the correct zero-differences.

As we can see, these differences move on to the right component after 4 rounds. There are similar trails covering it described in the appendix (Figure 18), the first is also represented on Figure 14b. As before, gray represents zero differences, black non-zero ones and black squares the cancellations which must occur during encryption. It also represents in dotted light blue the difference propagation during the first 3 rounds without any constraints regarding the cancellations so that this trail has probability 1. The green squares represent the cancellations which must be observed when starting from the bottom and partially decrypting a pair of ciphertext having the correct output difference.

It is therefore possible to cover as many rounds as we want using a characteristic $\mathcal{L}_i, \mathcal{R}_i, ..., \mathcal{L}_i, \mathcal{R}_i$ for any $i \in [1, 4]$. Such a trail would cover $4r$ rounds with probability $2^{-16 \cdot r}$. We also denote $\mathcal{L}'_i$ the trail $\mathcal{L}_i$ extended on top so as to have 8 non-zero input differences at the cost of one additional cancellation and $\mathcal{R}'_i$ the trail $\mathcal{R}_i$ reduced to 3 rounds and where no cancellations occur. Both $\mathcal{L}'_i$ and $\mathcal{R}'_i$ correspond to the case where the dotted light blue lines contain non-zero differences.

5.2 Efficient Key Recovery

The 4 cancellations (5 during the very first round) preventing the difference from spreading to the other component can be grouped into 2 sets each depending on a distinct set of 5 and 6 sub-keys. This phenomenon is illustrated on Figure 15 where zero differences are in gray, the first sub-component is represented with a continuous line and the second with a dashed line. The cancellation during the first round of the iterated trail is only relevant during the very first round of encryption.

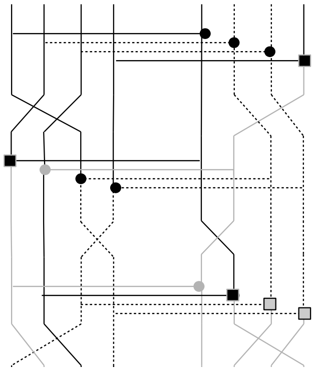


Fig. 15: Which S-boxes and sub-keys are involved in the 5 cancellations happening in $\mathcal{L}'_1$. Grey lines correspond to zero differences, squares to cancellations.

Starting from a pair of plaintexts separated by the correct input difference, it is easy to generate the set of all the sub-keys combinations which would lead to the trail we expect as follows:

1. Try all possible combinations of the sub-keys involved in the continuous (i.e. "not dashed") part of Figure 15 and store only those leading to the correct cancellations. There are $2^{4 \cdot 5} = 2^{20}$ possibilities, out of which $2^{20-3 \cdot 4} = 2^8$ lead to the correct pattern.
2. Try all possible combinations of the sub-keys involved in the dashed part of Figure 15 and store only those leading to the correct cancellations. There are $2^{4 \cdot 6} = 2^{24}$ possibilities, out of which $2^{20-2 \cdot 4} = 2^{16}$ lead to the correct pattern.
3. Combine the 2^8 and 2^{16} independent sub-candidates to obtain 2^{24} candidates of $4 \cdot (5+6) = 44$ bits each.

A very similar algorithm can be used to recover the candidates yielding the correct cancellations when partially decrypting the ciphertexts of the same pair. Doing so generates another 2^{24} candidates of 44 bits each.

5.3 Combining Truncated Differentials to Attack 23-Round TWINE-128

General Principle The high level idea of this attack is to discard some combinations of values for the set made of the 12 sub-keys used to update the left component during the first 3 rounds and the 12 sub-keys used to update the right component during the last 3 rounds. These form of set of 24 nibbles, i.e. 96 bits. The first and last 4-round blocks of the truncated differential trails described in Figure 16 all depend on the sub-keys in this set, although each of the trails only uses a different set of 88 bits out of the 96 bits available. It is therefore easy to combine the information deduced from each. A complete description of our attack follows.

Using the trails described in the previous Section, we can cover 23 rounds with probability $p = 2^{-84}$ in four different ways. The chaining of these different 4-round characteristics is described in Figure 16 where a 0 means there is no difference on this nibble and a x means any non-zero difference. Note that they all require the same input truncated difference, all yield the same output truncated difference and once a branch has been "selected" during the third round by cancelling one of the difference, the truncated trail is fixed.

1. **Data generation** First of all, we need to generate the pairs from which we are going to extract information about the sub-keys. For this purpose, we use 2^s structures of 2^{32} plaintext/ciphertext couples each. In these structures, nibbles $x_0[0..3, 6, 7, 10, 11]$ take all possible values while the others are constant. We thus obtain 2^{s+63} pairs with the correct input difference at a cost of $D = 2^{32+s}$ queries to an encryption oracle. We then obtain all the pairs which also have the correct output difference, namely[2] 0^8x^8, at the cost of 2^s sorting of arrays of 2^{32} ciphertexts. Since this output difference has probability $f = 2^{-32}$, this leaves $N_p = 2^{s+63} \cdot f = 2^{s+31}$ pairs with the correct input and output differences. Among these, there are $N_r = 2^{s+63} \cdot p = 2^{s-21}$ right pairs for each of the 4 truncated differential trails described in Figure 16 — which means that s must be at least equal to 21. Note that $N_p = N_r f/p$ and $D = N_r/p$.
   Now that we have the data we need, we process as follows for each of the 4 trails, t being the index of the trail considered.
2. **Counters increment** For $t \in [1, 4]$
   (a) Let $\mathcal{T}_t$ be an array of size 2^{88}. For each of the N_p pairs which passed the filter, we run the algorithms described in Section 5.2 to recover 2^{24} sub-candidates for the subset of 11 sub-keys used in the first 3 rounds and 2^{24} sub-candidates for the other subset of 11 sub-keys used in the last 3 rounds. This leads to $K = 2^{48}$ candidates living in space of size $\mathcal{S} = 2^{88}$
3. **Discarding candidates** We now have 4 tables $\mathcal{T}_t, t \in [1, 4]$ of $\mathcal{S}$ counters. In each table, each of the $\mathcal{S}$ candidates has been N_p times incremented with probability $K/\mathcal{S} = 2^{-40}$. We thus approximate the distribution of the counters by a normal distribution with average value $\mu_{\text{wrong}} = N_p K/\mathcal{S} = N_r(fK)/(p\mathcal{S})$ and variance $\sigma^2_{\text{wrong}} = N_p(K/\mathcal{S})(1 - K/\mathcal{S}) \approx N_r(fK)/(p\mathcal{S})$. However, the correct counter has also been incremented by each of the N_r correct pairs, meaning that its average value is $\mu_{\text{right}} = N_p K/\mathcal{S} + N_r =$

[2] The order of the nibbles in this difference corresponds to the order of the nibbles in our alternative representation.

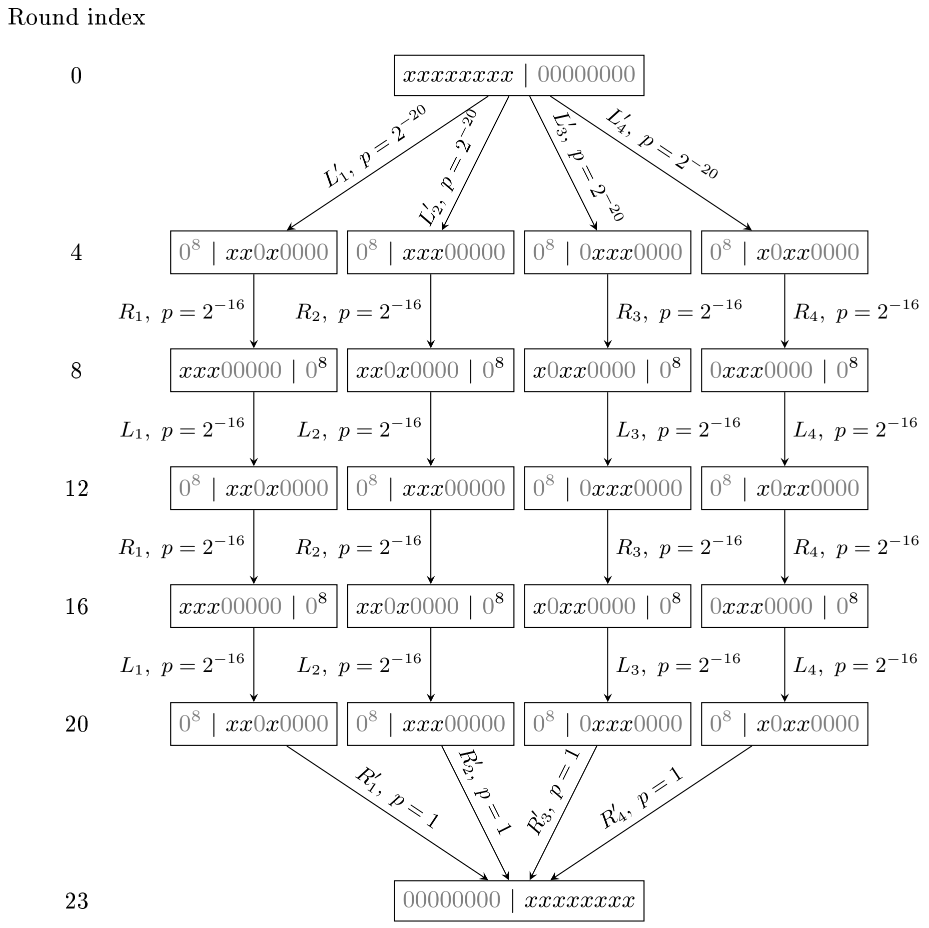


Fig. 16: The four distinct 23-round truncated differential trails we use to attack TWINE. The nibbles are ordered as in the left side of Figure 12.

$N_r\big((fK)/(p\mathcal{S})+1\big)$. We define μ_0 in order to express μ_{wrong}, σ^2_{wrong} and μ_{right} easily:

$$\mu_0 = \frac{f \cdot K}{p \cdot \mathcal{S}},\ \mu_{\text{wrong}} = N_r\mu_0,\ \sigma^2_{\text{wrong}} = N_r\mu_0,\ \mu_{\text{right}} = N_r(\mu_0+1).$$

We then combine the information from these counters. To achieve this, we recall that the indices in the tables $\mathcal{T}_t$ correspond to different subsets of 88 bits of a set of sub-keys of 96 bits in total. Therefore, we can associate a single representative in each table $\mathcal{T}_t$ to each candidate of 96 bits. Hence, we can give a score to each 96-bits candidate by taking the average of the scores of their representatives in each table. As a consequence, the score of a wrong candidate follows a normal distribution with the following parameters:

$$\mathcal{N}\big(\frac{4\cdot\mu_{\text{wrong}}}{4},\frac{4\cdot\sigma^2_{\text{wrong}}}{4^2}\big) \;=\; \mathcal{N}\big(\mu_{\text{wrong}},\frac{\sigma^2_{\text{wrong}}}{4}\big).$$

Similarly, the score of the right candidate is a sample from a distribution $\mathcal{N}\big(\mu_{\text{right}},\sigma^2_{\text{right}}/4\big)$. If we want a probability of keeping the right candidate of about $1/2$, we need to discard all the candidates having a score below μ_{right}. We denote $\mathbb{P}_{\text{wrong}}$ the probability to keep a wrong candidate, i.e. the probability that a wrong candidate has a score greater than μ_{right}. It is given by:

$$\mathbb{P}_{\text{wrong}} = \frac{1}{2}\Big[1-\text{erf}\Big(\frac{\mu_{\text{right}}-\mu_{wrong}}{\sqrt{2\sigma^2_{\text{wrong}}/4}}\Big)\Big] = \frac{1}{2}\Big[1-\text{erf}\Big(\sqrt{\frac{2N_r}{\mu_0}}\Big)\Big]. \qquad (1)$$

As we can see and unsurprisingly, the amount of wrong candidates discarded increases with the number N_r of right pairs for each trail. Table 2 gives the value of the probability $\mathbb{P}_{\text{wrong}}$ to keep a wrong candidate depending on the value of N_r as well as the corresponding data complexity knowing that $\mu_0 = 2^{-32+56+84-96} = 2^{12}$. Note also that the maximum value of N_r corresponds to the full code-book, i.e. when we query all 2^{32} possible structures, in which case, $N_r = 2^{32-21} = 2^{11}$.

Complexity Estimation The memory complexity of the truncated differential attack described in the previous section is straight-forward to evaluate. We need to store at most 2^{63} plaintext/ciphertext pairs and 4 times 2^{88} counters. These counters are on average equal to $N_r \cdot 2^{12}$ with N_r equal to at most 2^{11}. Hence, 32 bits are more than enough for each of them. Storing the counters is clearly the dominating factor here, meaning that the memory complexity of this attack is $4\cdot 2^{88} = 2^{90}$ counters of 32 bits or 2^{89} internal states.

We need $N_r \cdot 2^{53}$ plaintext/ciphertext pairs, meaning that the data complexity is $N_r \cdot 2^{53}$.

This also implies that we need at least the time taken to generate these. Furthermore, we also need to compute the possible candidates for each of the $N_r \cdot 2^{52}$ pairs which passed the filter. As seen in Section 5.2, this can be done in time 2^{48} for each pair. Hence, we also need to perform a counter increment $4\cdot N_r \cdot 2^{52}\cdot 2^{48} = N_r \cdot 2^{102}$ times. Finally, for all the candidates with a high enough score, we need to brute-force the 32 remaining bits of the key. This requires $2^{128}\cdot\mathbb{P}_{\text{wrong}}$ encryptions. The complexities for different values of N_r are given in Table 2.

6 Optimizing the Search for High Probability Differentials

While truncated differentials can be used directly to attack (round-reduced) block ciphers directly, they can also be used to optimize the search for high

N_r	$\mathbb{P}_{\text{wrong}}$	D	T	M
2^{5}	$2^{-1.22}$	2^{58}	$2^{126.78}$	2^{89}
2^{7}	$2^{-1.47}$	2^{60}	$2^{126.53}$	
2^{9}	$2^{-2.06}$	2^{62}	$2^{125.94}$	
2^{11}	$2^{-3.67}$	2^{64}	$2^{124.34}$	

Table 2: Data, time and memory complexity of a truncated differential attack on TWINE-128.

probability differentials. Indeed, by providing a "template" which differential characteristics should follow, it can reduce the size of the search space significantly and make the computation of a lower bound on a differential probability tighter. A similar approach was used in [18] to identify high probability differentials for PRINCE which were then used in a multiple differential attack which is the best attack on this cipher today. LAC [6], a lightweight candidate of the CAESAR competition based on a simplified version of LBlock called LBlock-s, has been the target of another high probability differential search in a note released online by Leurent [19].

In both cases, the method has been the same: first identify a high probability differential trail and then use a heuristic method to compute a lower bound on the probability of a differential by essentially clustering all characteristics following said truncated differential. Since we have iterated truncated trails covering any amount of rounds for TWINE, we apply this method on this cipher to identify high probability differentials.

For a truncated characteristic T covering r rounds, we denote $P_T[\delta \rightarrow \Delta]$ the probability of the differential $(\delta \rightarrow \Delta)$ obtained by summing the probabilities of all the differential trails mapping δ to Δ which follow the truncated trail. Using these probabilities, we build a matrix $M(C)$ such that $M(T)_{i,j} = P_T[i \rightarrow j]$. To obtain the distribution of Δ given δ, we simply multiply a vector made of zeroes everywhere except in position δ, where it is equal to 1, by $M(T)$. Note that the sum of the probabilities of the Δ's obtained in this fashion is not equal to 1 as the truncated trail itself does not have a probability of 1. Given $M(T)$, finding the differential with the highest probability can be done easily by finding the maximum coefficient in the matrix. The size of $M(T)$ is limited by only taking into account the values of δ and Δ which are coeherent with T.

In order to obtain the distribution of Δ after two iterations of the trail T, we multiply the same vector by the matrix $M(T) \times M(T)$, where "×" denotes regular matrix multiplication. This construction can of course be iterated.

In the case of TWINE, we computed two matrices $M(\mathcal{L}_1)$ and $M(\mathcal{R}_1)$ corresponding to the truncated trails $\mathcal{L}_1$ and $\mathcal{R}_1$ described in Figures 14a and 14b respectively. Both $M(\mathcal{L}_1)$ and $M(\mathcal{R}_1)$ are square matrices of size $2^{12} \times 2^{12}$ because both trails have only 3 non-zero nibbles as both their input and output.

Using different multiplications of these, we found the high probability differentials given in Table 3.

Rounds	Input difference	Output difference	Probability	# Active S-Boxes $\times 2^{-2}$
4	10 20 00 60 00 00 00 00	00 00 20 00 60 00 00 60	$2^{-17.496}$	2^{-18}
	60 20 00 60 00 00 00 00	00 00 20 00 60 00 00 10	$2^{-17.496}$	
	30 60 00 30 00 00 00 00	00 00 60 00 30 00 00 10	$2^{-17.759}$	
	10 60 00 30 00 00 00 00	00 00 60 00 30 00 00 30	$2^{-17.759}$	
8	10 20 00 60 00 00 00 00	60 20 00 10 00 00 00 00	$2^{-34.542}$	2^{-36}
	10 20 00 60 00 00 00 00	60 20 00 f0 00 00 00 00	$2^{-34.981}$	
	f0 20 00 60 00 00 00 00	60 20 00 10 00 00 00 00	$2^{-34.981}$	
	d0 f0 00 80 00 00 00 00	80 f0 00 d0 00 00 00 00	$2^{-34.994}$	
12	10 20 00 10 00 00 00 00	00 00 20 00 60 00 00 10	$2^{-52.083}$	2^{-54}
	10 20 00 60 00 00 00 00	00 00 20 00 10 00 00 10	$2^{-52.083}$	
	80 f0 00 80 00 00 00 00	00 00 f0 00 d0 00 00 80	$2^{-52.144}$	
	80 f0 00 d0 00 00 00 00	00 00 f0 00 80 00 00 80	$2^{-52.144}$	
16	60 20 00 60 00 00 00 00	60 20 00 60 00 00 00 00	$2^{-67.538}$	2^{-72}
	30 60 00 30 00 00 00 00	30 60 00 30 00 00 00 00	$2^{-67.595}$	
	90 30 00 90 00 00 00 00	90 30 00 90 00 00 00 00	$2^{-67.626}$	
	80 f0 00 80 00 00 00 00	80 f0 00 80 00 00 00 00	$2^{-67.762}$	

Table 3: High probability differentials for round-reduced TWINE.

As we can see, the highest probability for a differential over 4 rounds is higher than we might expect. Indeed, 9 S-Boxes are involved in it and the maximum probability for a differential in the S-Box is 2^{-2}. Hence, the maximum probability of a characteristic is 2^{-18}, which is smaller than the value of $2^{-17.5}$ our model predicts and which we checked experimentally. The gain then increases as the number of rounds increases. For 12 rounds, we have 27 active S-Boxes which means that the probability of a characteristic cannot be higher than 2^{-54} and yet the highest differential probability is at least $2^{-52.1}$.

Leurent obtained more impressive results for LBlock-s (e.g. a lower bound of $2^{-29.8}$ for 8 rounds) which might be surprising at first glance since the linear layer of these two ciphers are very similar and both use S-Boxes with a maximum differential probability equal to 2^{-2}. However, the distribution of the coefficients in the difference distribution tables of the S-Boxes of these ciphers are different. For instance, with S_L and S_T denoting the S-Boxes of LBlock-s and TWINE respectively, we have $P\big[S_L(x+\delta)+S_L(x)=4\big]=2^{-2}$ for $\delta \in \{4,5,6,7\}$ while there exists only one δ such that $P\big[S_T(x+\delta)+S_T(x)=\varDelta\big]=2^{-2}$ for any $\varDelta \neq 0$. In other words, the distribution of the output differences is closer to being uniform in TWINE than in LBlock-s (and LBlock). To study the consequences of these variation in differential behaviour, we reiterated our differential search by replacing the S-box of TWINE by that of LBlock-s. We obtained four distinct differentials with probability at least $2^{-31.7}$ for 8 rounds.[3] This result is $2^{4.3}$ times better than what a wide-trail argument would give and 2^3 times higher than for the TWINE S-Box.

Our findings highlight both how large truncated differentials can be leveraged to prove tighter lower bounds on differential probabilities and how the distribution of the coefficients in the difference distribution table of a S-Box as a whole should be taken into account when designing a primitive in contrast to simply looking at the maximum coefficient, as is often the case when wide-trail arguments are used. For $n \times n$ S-Boxes affine equivalent to monomials of $GF(2^n)$, this distribution is fully described by the so-called *differential spectrum* [26] but, to the best of our knowledge, there is no generalization of this concept to arbitrary S-Boxes.

7 Conclusion

Suzaki et al. proposed a new type of permutation to be used in GFN's in [3] and later applied it to design TWINE. We presented two new attacks on 25 rounds out of 36 of this primitive which are, to the best of our knowledge, the best attacks in the single-key model. We then shed new light on the way information propagates in such a modified GFN and showed that the mixing actually operates in two phases: two halves of the internal state are mixed independently for three rounds and only exchange information during the fourth round. This behaviour is repeated *ad infinitum* and can also be observed in LBlock and its variant, LBlock-s. We used this observation to find high probability truncated differential trails and then leveraged these results to both attack 23-rounds TWINE-128 and give a tighter lower bound on the high probability of some differentials, highlighting differences between TWINE and LBlock-s with regards to differential propagation in the process.

References

1. Biryukov, A., Perrin, L.: State of the art in lightweight cryptography. http://cryptolux.org/index.php/Lightweight_Cryptography
2. Nyberg, K.: Generalized feistel networks. In Kim, K., Matsumoto, T., eds.: Advances in Cryptology – ASIACRYPT '96. Volume 1163 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1996) 91–104
3. Suzaki, T., Minematsu, K.: Improving the generalized feistel. In: Fast Software Encryption, Springer (2010) 19–39
4. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A Lightweight Block Cipher for Multiple Platforms. In: Selected Areas in Cryptography, Springer (2013) 339–354

[3] Note that Leurent used a truncated differential with 17 active S-Boxes while our has 18. This difference is likely to account for the factor $2^{1.9}$ separating our results.

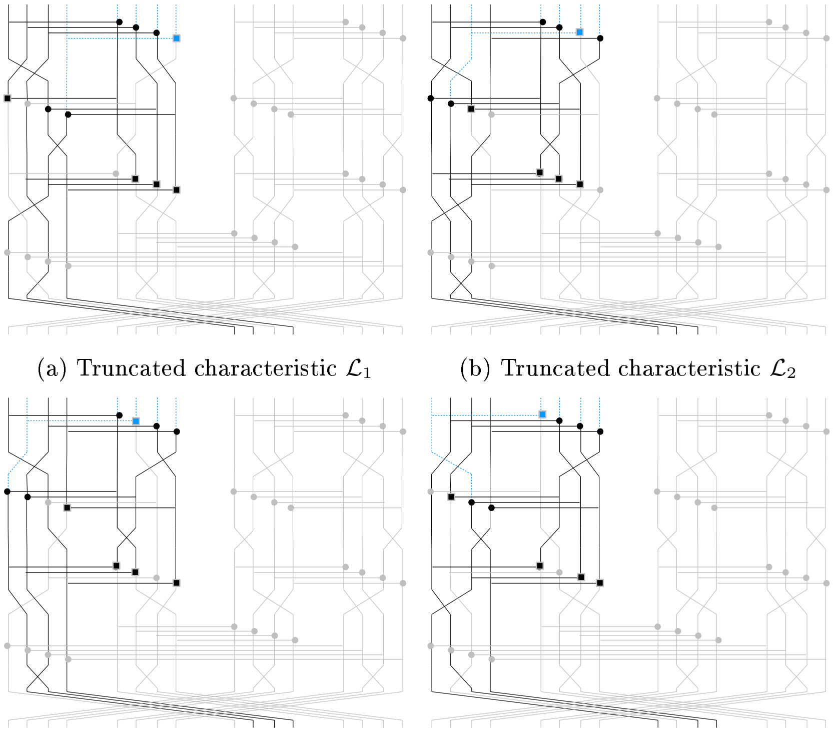
5. Wu, W., Zhang, L.: Lblock: a lightweight block cipher. In: Applied Cryptography and Network Security, Springer (2011) 327–344
6. Zhang, L., Wu, W., Wang, Y., Wu, S., Zhang, J.: Lac: A lightweight authenticated encryption cipher. Candidate for the CAESAR Competition (2014)
7. Boztas, Ö., Karakoç, F., Çoban, M.: Multidimensional meet-in-the-middle attacks on reduced-round twine-128. In: LightSec. (2013) 55–67
8. Wen, L., Wang, M., Bogdanov, A., Chen, H.: Note of multidimensional mitm attack on 25-round twine-128. Cryptology ePrint Archive, Report 2014/425 (2014) http://eprint.iacr.org/.
9. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Fast Software Encryption, Springer (2008) 116–126
10. Derbez, P., Fouque, P.A., Jean, J.: Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In: EUROCRYPT. (2013) 371–387
11. Derbez, P., Fouque, P.: Exhausting Demirci-Selçuk meet-in-the-middle attacks against reduced-round AES. In: Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. (2013) 541–560
12. Li, L., Jia, K., Wang, X.: Improved meet-in-the-middle attacks on aes-192 and prince. Cryptology ePrint Archive, Report 2013/573 (2013) http://eprint.iacr.org/.
13. Boura, C., Naya-Plasencia, M., Suder, V.: Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. (2014) 179–199
14. Wang, Y., Wu, W.: Improved multidimensional zero-correlation linear cryptanalysis and applications to lblock and TWINE. In: Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings. (2014) 1–16
15. Knudsen, L.R.: Truncated and higher order differentials. In: Fast Software Encryption, Springer (1995) 196–211
16. Lallemand, V., Naya-Plasencia, M.: Cryptanalysis of KLEIN. In: Fast Software Encryption. Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) To appear
17. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: a new family of lightweight block ciphers. In: RFID. Security and Privacy. Springer (2012) 1–18
18. Canteaut, A., Fuhr, T., Gilbert, H., Naya-Plasencia, M., Reinhard, J.R.: Multiple differential cryptanalysis of round-reduced prince (full version). Cryptology ePrint Archive, Report 2014/089 (2014) http://eprint.iacr.org/.
19. Leurent, G.: Differential Forgery Attack against LAC. https://hal.inria.fr/hal-01017048 (July 2014)
20. Çoban, M., Karakoç, F., Boztaş, Ö.: Biclique cryptanalysis of twine. In: Cryptology and Network Security. Springer (2012) 43–55
21. Zheng, X., Jia, K.: Impossible differential on reduced-round TWINE. In: International Conference on Information Security and Cryptology. (2014) To appear
22. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems-CHES 2007. Springer (2007) 450–466

23. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. In: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. (2010) 158–176
24. Knudsen, L.R.: Deal – a 128-bit block cipher. Technical Report Department of Informatics (1998)
25. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. (1999) 12–23
26. Blondeau, C., Canteaut, A., Charpin, P.: Differential properties of power functions. International Journal of Information and Coding Theory **1**(2) (2010) 149–170

A Appendix

A.1 Complete 4-Rounds Truncated Differential Characteristics for TWINE

In this Section, we present all the 4-rounds truncated differential trails we use to attack TWINE. Figure 17 describes trails on the left component and how they can be extended, at the cost of an additional cancellation, to have a larger input difference. Figure 18 describes trails on the right component and how they behave during the first 3 rounds if no cancellation occur.



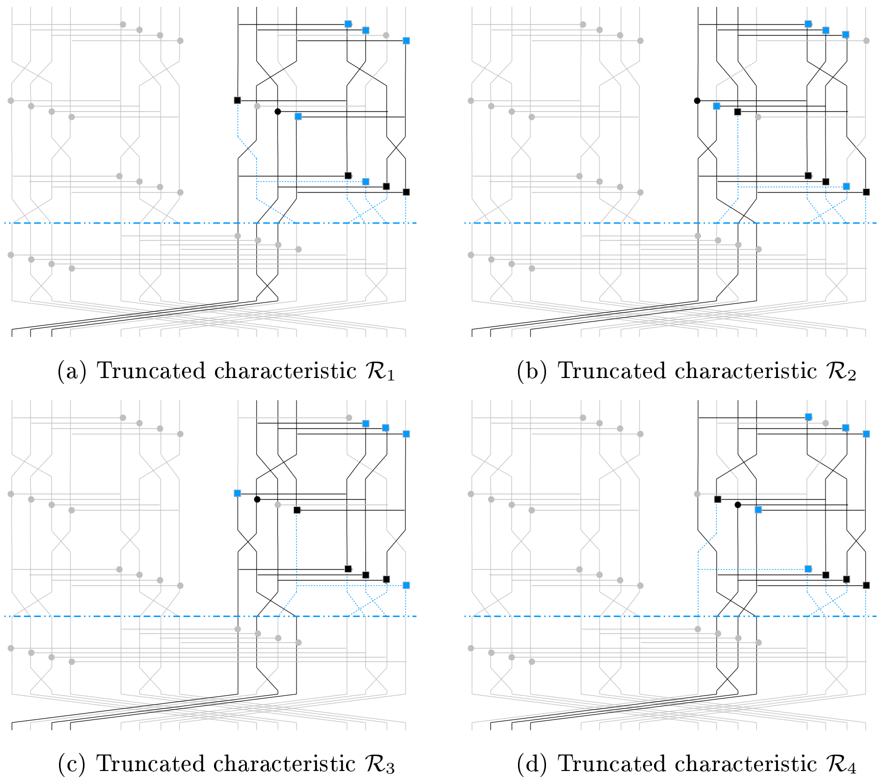
(a) Truncated characteristic $\mathcal{L}_1$

(b) Truncated characteristic $\mathcal{L}_2$

(c) Truncated characteristic $\mathcal{L}_3$

(d) Truncated characteristic $\mathcal{L}_4$

Fig. 17: Truncated differential characteristics on the left component of TWINE and their extensions towards the top. Zero differences are represented in black and squares correspond to places where cancellations are necessary.



(a) Truncated characteristic $\mathcal{R}_1$

(b) Truncated characteristic $\mathcal{R}_2$

(c) Truncated characteristic $\mathcal{R}_3$

(d) Truncated characteristic $\mathcal{R}_4$

Fig. 18: Truncated differential characteristics on the right component of TWINE and their extensions towards the bottom.