Multiple differential cryptanalysis of round-reduced PRINCE [⋆]

Anne Canteaut[1], Thomas Fuhr[2], Henri Gilbert[2],
María Naya-Plasencia[1], and Jean-René Reinhard[2]

[1] Inria, France
[2] ANSSI, France

Abstract. PRINCE is a lightweight block cipher proposed by Borghoff *et al.* at Asiacrypt 2012. Due to its originality, novel design and low number of rounds, it has already attracted the attention of a large number of cryptanalysts. Several results on reduced versions have been published to date; the best one is an attack on 8 rounds out of the total number of 12. In this paper we improve this result by two rounds: we provide an attack on 10 rounds of the cipher with a data complexity of $2^{57.94}$ and a time complexity of $2^{60.62}$, corresponding to 118.56 security bits, instead of 126 for the generic attacks. Our attack uses multiple differentials and exploits some properties of PRINCE for recovering the whole key. PRINCE is defined as a member of a family of ciphers, differing by the choice of an Sbox among a distinguished set. We also show that the security offered by all the members of the family is not equivalent, by identifying an Sbox for which our attack can be extended up to 11 rounds with a data complexity of $2^{59.81}$ and a time complexity of $2^{62.43}$.

Keywords. Differential cryptanalysis, PRINCE, multiple differentials, key-recovery.

1 Introduction

The area of lightweight primitives has drawn considerable attention over the last years, due to the need of low-cost cryptosystems for several emerging applications like RFID tags and sensor networks. The strong demand from industry has led to the design of a large number of lightweight block ciphers, with different implementation features. Among the best studied proposals are the ISO/IEC standards PRESENT [4] and CLEFIA [16], as well as LBlock [19], TWINE [18], LED [12] and KLEIN [11]. In this context, the need for a significant cryptanalysis effort is obvious. The demand from industry for clearly recommended lightweight ciphers requires that the large number of these potential candidates be narrowed down. Since the trade-off between the performance and the security is a major issue for lightweight primitives, it is also very important to estimate the security

[⋆] Partially supported by the French Agence Nationale de la Recherche through the BLOC project under Contract ANR-11-INS-011.

margin of these ciphers, to determine for instance if some rounds need to be added, or if some can be omitted for achieving a given security level.

Recently, at Asiacrypt 2012, a new lightweight cipher named PRINCE has been proposed by Borghoff *et al.* [5]. This block cipher, which aims at low-latency encryption, has already received a lot of attention from the community, mainly due to its simplicity, originality and low number of rounds. These results include some small improvements upon the generic attack against the full cipher (that implies a reduction from 2^{127} to 2^{126} of the claimed security bound on the product $\mathsf{Data} \times \mathsf{Time}$ of the data and time complexities for any attack) [13, 1]. Against round-reduced versions of PRINCE, the best attack published so far applies to 8 rounds (out of 12) [8] and an attack against 9 rounds has been described very recently [14]. Also some analysis of the building block PRINCE_{core} and some interesting results have been obtained on a variant of PRINCE using a chosen weak constant α [17, 13]. However, this constant is not a parameter of the design of PRINCE. An attack in the multi-user setting has also been presented in [9].

In this paper, we propose a differential-type attack on round-reduced PRINCE that increases the number of analyzed rounds to 10 rounds, without modifying the constants or building-blocks in the cipher. It is a multiple differential attack, based on a principle similar to the one in [15], that we have combined with a sophisticated key recovery method specific for PRINCE. The fact that the linear layer in PRINCE is based on the same design strategy as the AES aims at making it resistant to classical differential attacks. In particular, due to the branch number of the linear transformation, any differential characteristic over four consecutive rounds has at least 16 active Sboxes, implying that the probability of any differential characteristic over the 12 rounds is at most 2^{-96}. Nevertheless, our attack exploits the following four properties coming from the main features of PRINCE:

- there exist many differentials for the round function with 4 active Sboxes and with an activity pattern having a particular shape (the active nibbles are the corners of a square);
- several of the characteristics obtained by iterating these round differentials have the same input and the same output differences, leading to some r-round differentials whose probability is much higher than the probability of a single characteristic;
- for a given pair of input and output activity patterns, we find several good differentials, which can be exploited together in a multiple differential attack, as proposed in [2, 3];
- because of the particular shape of the activity patterns, these differentials can be extended by two rounds in each direction. Indeed, for some fixed input and output activity patterns, the active nibbles only depend on half of the bits of the plaintext and of the ciphertext, and on 66 of the 128 key bits.

Altogether, these four properties enable us to describe the first attack on 10-round PRINCE, which requires $2^{57.94}$ chosen plaintexts, with time complexity less that $2^{60.61}$ encryptions, leading to a product $\mathsf{Data} \times \mathsf{Time}$ around $2^{118.56}$.

Another interesting issue is that, besides PRINCE, the designers have proposed a whole family of ciphers, named the PRINCE-Family, which differ in their 4×4 Sbox only. Since the Sbox of any member in the PRINCE-Family guarantees the same resistance to classical attacks, including differential attacks, all those ciphers seem to offer a similar security. Here, we show that it is not the case since all these Sboxes do not have the same behaviour regarding our attack. In particular, we exhibit a member of the PRINCE-Family for which up to 11 rounds can be attacked with data and time complexity satisfying $\mathsf{Data} \times \mathsf{Time} = 2^{122.24}$. The complexities of our attacks and a comparison with the previous results are given in Table 1.

Part of PRINCE	Source	Rounds	**D**ata	**T**ime	D×T	Memory	Attacks
PRINCE	[13]	6	2^{16}	2^{64}	2^{80}	2^{16}	integral
	[8]	8	1	2^{123}	2^{123}	2^{20}	sieve-in-the-middle
	[14]	9	2^{57}	2^{64}	2^{121}	$2^{57.3}$	meet-in-the-middle
	Sec. 6	9	$2^{46.89}$	$2^{51.21}$	$2^{98.10}$	$2^{52.21}$	multiple differential[3]
	Sec. 6	10	$2^{57.94}$	$2^{60.62}$	$2^{118.56}$	$2^{61.52}$	multiple differential[3]
PRINCE(chosen α)	[13]	12	2^{41}	2^{41}	2^{82}	-	boomerang
PRINCE$_{core}$	[13]	6	2^{16}	2^{30}	2^{46}	2^{16}	integral
	[1]	12	2^{40}	$2^{62.72}$	$2^{102.72}$	2^{8}	biclique
PRINCE-Family (modified Sbox)	Sec. 6	10	$2^{50.42}$	$2^{53.61}$	$2^{104.03}$	$2^{54.00}$	multiple differential[3]
	Sec. 6	11	$2^{59.81}$	$2^{62.43}$	$2^{122.24}$	$2^{63.39}$	multiple differential[3]

Table 1. Summary of all attacks on PRINCE in the single-key setting, including the attacks presented in this paper.

The paper is organized as follows. After a description of PRINCE in Section 2, Section 3 exhibits some differential paths with 4 active nibbles per round only, and gives a lower bound on the probabilities of some related differentials. Section 4 shows how these r-round differentials can be extended by two rounds at the beginning and by two rounds at the end, in a way such that some key bits can be recovered. Some experimental results supporting the previously made assumptions are presented in Section 5. Section 6 discusses and presents the results on 9 and 10 rounds of PRINCE and of some other element in the PRINCE-Family.

[3] Memory complexity figures can be slightly larger than time complexities due to the relative cost between an encryption and a memory access.

2 The PRINCE block cipher

PRINCE operates on 64-bit blocks and uses a 128-bit key composed of two 64-bit elements, K_0 and K_1. Its structure is depicted on Figure 1.

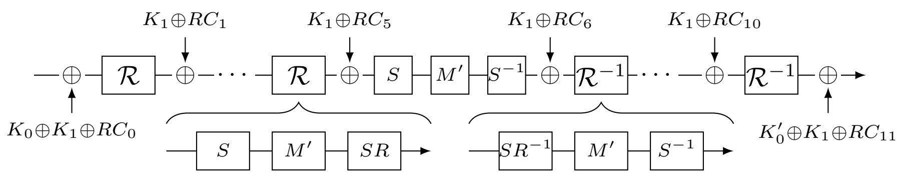

Fig. 1. Structure of PRINCE.

PRINCE is based on the so-called FX-construction: two whitening keys $W_{in} = (K_0 \oplus K_1)$ and $W_{out} = (K'_0 \oplus K_1)$ are XORed respectively to the input and to the output of a 12-round core cipher, named PRINCE$_{core}$, parametrized by K_1 only. The value of K'_0 involved in the post-whitening key is derived from K_0 by $K'_0 = (K_0 \ggg 1) \oplus (K_0 \gg 63)$.

The internal state of the cipher is represented by a 4×16 binary matrix. The lower right corner of this matrix is the least significant bit of the input, while the upper left corner is the bit of index 63. This 4×16 binary matrix can also be seen as a 4×4 matrix of nibbles. In this case, the nibbles are numbered by their positions in the matrix, where the rows (resp. the columns) are numbered from top to bottom (resp. from left to right). The precise numbering of bit positions and nibbles is depicted on Figure 2. We adopt the following notation. For a 64-bit state, plaintext or ciphertext value W, when W is seen as a matrix of nibbles, W^j designates column j of W, and $W^{i,j}$ designates the nibble at row i and column j. W_i designates the bit at position i. For a 64-bit key value K, K^i designates the bit of K at position i. For a general 64-bit value W and a set of bit indexes E, W^E designates the set of bits of W at bit positions in the set E.

	Column 0				Column 1				Column 2				Column 3			
Row 0	63	62	61	60	47	46	45	44	31	30	29	28	15	14	13	12
Row 1	59	58	57	56	43	42	41	40	27	26	25	24	11	10	9	8
Row 2	55	54	53	52	39	38	37	36	23	22	21	20	7	6	5	4
Row 3	51	50	49	48	35	34	33	32	19	18	17	16	3	2	1	0

Bits numbering

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Nibbles numbering

Fig. 2. Numbering of nibbles and bits of the internal state.

The round function is then composed of:

- a non-linear layer S corresponding to 16 parallel applications of a 4×4 Sbox σ, which operates on the 16 nibbles of the internal state.
- a linear layer $SR \circ M'$, where M' is the parallel application of 4 involutive MixColumn transformations, each operating on 16 bits. The same transformation is applied on first and last (resp. on second and third) columns of the state. This transformation is then followed by a permutation SR of the 16 nibbles which is similar to the AES ShiftRows operation: in the 4×4 matrix of nibbles, the row of index i is rotated by i positions to the left.
- the addition of a round constant RC_i and of the subkey K_1.

The first 5 rounds in PRINCE correspond to the previously described round permutation $\mathcal{R}$, while the last 5 rounds are defined by the inverse permutation $\mathcal{R}^{-1}$. The middle round corresponds to the successive applications of S, M' and S^{-1} (see Figure 1).

One of the main features of PRINCE$_{core}$ is that decryption can be implemented by the same circuit as encryption, but under another key. This comes from the fact that the 12 round constants satisfy $RC_i \oplus RC_{11-i} = \alpha$ for $0 \leq i \leq 11$, where α is a fixed constant, implying that decryption under key K_1 corresponds to encryption under key $K_1 \oplus \alpha$.

Round-reduced versions of PRINCE. The number of rounds in PRINCE corresponds to the number of nonlinear layers. There are 12 rounds for the full cipher. Round-reduced versions are defined in a similar way: if the overall number of rounds r is even, the numbers of rounds before and after the middle transformation are the same, while they differ by 1 when r is odd.

Some observations on the linear layer. We note that M' can be expressed as the parallel application of 16 independent transformations operating on one column of bits of the internal state. These 4-bit transformations have the following form:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \mapsto \begin{pmatrix} wt(x) \bmod 2 \\ wt(x) \bmod 2 \\ wt(x) \bmod 2 \\ wt(x) \bmod 2 \end{pmatrix} \oplus \mathsf{Rot}_{n_i} \begin{pmatrix} x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} ,$$

where $wt(x)$ is the Hamming weight of x and Rot_{n_i} denotes some rotation by n_i positions to the top. In other words, M' is the composition of three simple columnwise operations: the inversion of the order of the four bits, followed by a bitwise rotation, completed by the addition of the parity of the column to each bit of the column. The whole linear layer when the weight of each column is even is depicted in the appendix of [7].

3 Differentials with 4 active nibbles for PRINCE$_{core}$

We now study some differentials for PRINCE$_{core}$ with particular activity patterns, and we compute a lower bound on their probabilities.

In the following, we denote by R the permutation corresponding to $S \circ SR \circ M'$. Evaluating the difference propagation through R enables to evaluate the difference propagation through one round $\mathcal{R}$ of PRINCE$_{core}$ since key and round constant additions do not alter the differences. To assess the probability of characteristics over several rounds, we will consider that the transition probabilities in different rounds are independent. Due to the absence of key addition around the middle M' layer, we apply a specific analysis to the surrounding non-linear layers. We denote by S_{sbox} the permutation $S^{-1} \circ M' \circ S$, which covers one Super Sbox, as defined in [10]. Here, we study some differentials for the function

$$\mathcal{F}_{r_1+r_2+2} = \left(R^{-1}\right)^{r_2} \circ M' \circ SR^{-1} \circ S_{sbox} \circ SR \circ M' \circ R^{r_1} ,$$

which should also be good differentials for $\mathcal{F}^{K_1}_{r_1+r_2+2}$, the function obtained by considering $(r_1 + r_2 + 2)$ rounds of PRINCE$_{core}$, with constants, key additions, and an additional linear layer at the beginning and at the end.

In the following, the internal state is seen as a 4×4 matrix of nibbles numbered according to Figure 2. Since any differential characteristic over four consecutive rounds has at least 16 active Sboxes [6, Theorem 2], we here focus on differential characteristics with four active Sboxes in each round. We show that several such characteristics can be built, and we additionally exhibit such characteristics sharing their input and output differences. In other words, we are able to find a lower bound on the probability of some *differentials* which include several differential characteristics with the lowest possible number of active Sboxes.

The crucial observation is that the diffusion in the linear layer is ensured by the addition of the parity in M'. In order to build some characteristics with a low number of active Sboxes, we focus on the differences with four active nibbles located in two columns, such that the two active nibbles in the same column are identical. This ensures that no diffusion of the differences takes place since the corresponding parity bits are inactive. We further restrict the differences considered to those whose active nibbles are the corners of a 3×3 *square*, *i.e.*, the positions of the four active nibbles in the 4×4 internal state are of the form (i, c), $(i+2, c)$, $(i, c+2)$ and $(i+2, c+2)$ for some i and c in $\{0, 1\}$. A difference satisfying these properties will be called *valid*.

There are exactly four square activity patterns and each of them is denoted by $[i, c]$ with the minimal values of i and c. In the following, any valid difference is then characterized by its activity pattern $[i, c]$ and by the two differences (δ_1, δ_2) where δ_1 (resp. δ_2) denotes the difference which appears in Column c (resp. in Column $(c + 2)$).

3.1 Computing the transition probabilities

There are exactly $15^2 = 225$ valid differences for each activity pattern. We now determine when a valid difference can be mapped into another valid difference by $R = S \circ SR \circ M'$. Since S operates on the nibbles independently, it does not affect the activity pattern. Then, we need to determine when a square activity pattern can be transformed into another square activity pattern by the linear layer $SR \circ M'$. It can be easily checked that this situation occurs if and only if the nibble differences (δ_1, δ_2) belong to the 18 pairs of the form

$$(\delta_1, \delta_2) \in (\Delta_1 \times \Delta_2) \cup (\Delta_2 \times \Delta_1) \text{ with } \Delta_1 = \{1, 4, 5\} \text{ and } \Delta_2 = \{2, 8, 10\} .$$

Then, there are exactly $4 \times 2 \times 9 = 72$ nonzero valid differences which are mapped by $SR \circ M'$ to a difference with a square activity pattern. The resulting differences have equal nibble differences on the square pattern diagonals. An example of such a propagation is depicted on Figure 3. Therefore, a valid input

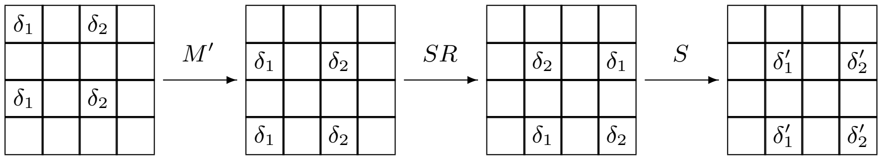

Fig. 3. Propagation of a valid input difference over one round R when $(\delta_1, \delta_2) \in \Delta_1 \times \Delta_2$ (if $(\delta_1, \delta_2) \in \Delta_2 \times \Delta_1$, the above sequence $[0, 0] \to [1, 0] \to [1, 1] \to [1, 1]$ of square activity patterns would be replaced by $[0, 0] \to [0, 0] \to [0, 0] \to [0, 0]$).

difference with active nibbles (δ_1, δ_2) can be mapped by R to a valid difference with active nibbles (δ'_1, δ'_2) if and only if the following four transitions for the Sbox are valid: $\delta_1 \mapsto \delta'_1$, $\delta_2 \mapsto \delta'_2$, $\delta_1 \mapsto \delta'_2$ and $\delta_2 \mapsto \delta'_1$. Then, the probability of the transition $(\delta_1, \delta_2) \mapsto (\delta'_1, \delta'_2)$ for R does not vary when the roles of δ_1 and δ_2 are inverted, or when the roles of δ'_1 and δ'_2 are inverted. This important property will be exploited in the attack. Thus, we consider the 9×9 transition matrix μ whose entry at the intersection of Row $(\delta_1, \delta_2) \in \Delta_1 \times \Delta_2$ and Column $(\delta'_1, \delta'_2) \in \Delta_1 \times \Delta_2$ is the product of the four probabilities:

$$p(\delta_1, \delta'_1)p(\delta_2, \delta'_1)p(\delta_1, \delta'_2)p(\delta_2, \delta'_2) ,$$

where

$$p(\delta, \delta') = \mathrm{Pr}_X[\sigma(X \oplus \delta) \oplus \sigma(X) = \delta']$$

and σ is the 4×4 Sbox used in PRINCE. Now, for a given input activity pattern, all 9 (δ_1, δ_2) in $\Delta_1 \times \Delta_2$ lead to the same output activity pattern, while all 9 (δ_1, δ_2) in $\Delta_2 \times \Delta_1$ lead to another one. Then, the whole 72×72 transition matrix can be written as the Kronecker product[4] $A \otimes \mu$, where the 8×8 matrix A encodes the transition between the 4 square activity patterns, together with the fact that δ belongs either to $\Delta_1 \times \Delta_2$ or to $\Delta_2 \times \Delta_1$, and the 9×9 matrix μ encodes transition between values of $\delta \in \Delta_1 \times \Delta_2$. The values of these two matrices are given in [7]. Thus, the transition matrix corresponding to r iterations of R is given by

$$(A \otimes \mu)^r = (A^r \otimes \mu^r) .$$

[4] The Kronecker product of an $m \times n$ matrix A and a $p \times q$ matrix B is the $mp \times nq$ block matrix whose block at Position (i, j) equals $A_{i,j}B$.

In the same way, we can define the matrices B and ν that correspond to the middle round $M' \circ SR^{-1} \circ S_{sbox} \circ SR \circ M'$. Note that due to the involutivity of S_{sbox}, ν is symmetric. Obviously, the transition matrix for R^{-1} is the transpose of the transition matrix for R, *i.e.*,

$$(A \otimes \mu)^T = (A^T \otimes \mu^T) \ .$$

It eventually follows that the transition matrix for $\mathcal{F}_{r_1+r_2+2}$ is

$$\left(A^{r_1} B (A^{r_2})^T \otimes \mu^{r_1} \nu (\mu^{r_2})^T\right) \ .$$

It can be checked that $AB = B(A)^T = J$, where J is the matrix with all entries equal to 1. Since all rows and all columns of A have weight 2, we deduce that $AJ = 2J$ and $JA^T = 2J$, implying that, for any $r_1 + r_2 \geq 1$,

$$A^{r_1} B (A^{r_2})^T = 2^{r_1+r_2-1} J \ .$$

It follows that, for any valid input difference defined by $\Delta_{in} = (\delta_1, \delta_2)$ and with any given input activity pattern, the probability that the output difference is a valid difference $\Delta_{out} = (\delta'_1, \delta'_2)$ with a given output activity pattern is

$$2^{r_1+r_2-1} \left(\mu^{r_1} \nu (\mu^{r_2})^T\right)_{i,j}$$

where i and j are the row and column indices corresponding to Δ_{in} and Δ_{out}. It is worth noticing that this probability depends on $(\Delta_{in}, \Delta_{out})$ only, and is independent from the input and output square activity patterns.

3.2 Results for the Sbox used in PRINCE

By computing $\mu^{r_1} \nu (\mu^{r_2})^T$, we get the following results for the Sbox used in PRINCE (the transition matrices are given in the appendix of [7]).

– For $r_1 = r_2 = 2$, i.e., for six rounds, the highest coefficient of the matrix $\mu^2 \nu (\mu^2)^T$ is $2^{-70} \times 1536$. Then, the best differential has probability

$$2^{-70} \times 1536 \times 2^3 \approx 2^{-56.42} \ .$$

This probability is obtained for four differentials, namely $(\Delta_{in}, \Delta_{out}) \in \{(1,2),(2,1)\} \times \{(1,2),(2,1)\}$, and for any fixed input and output activity patterns.

– For $r_1 = 1$ and $r_2 = 2$, the probability of the best differential is $2^{-47.42}$. For $r_1 = 2$ and $r_2 = 1$, the transition matrix $\mu^2 \nu (\mu)^T$ is obviously the transpose of the previous one, leading to the same maximal transition probability.

3.3 Results for another Sbox used in the PRINCE-Family

The PRINCE-family consists of all ciphers defined as PRINCE but with an Sbox σ which can be chosen among eight 4×4 Sboxes and all the affinely equivalent transformations (see Appendix B in [6]). All these Sboxes are expected to have the same properties regarding classical differential attacks. However, when we focus on the valid differentials with square activity patterns, these Sboxes do not have the same behaviour. We have searched for some Sbox minimizing the complexity of the attack described in the next section. An optimal Sbox in that sense, linearly equivalent to Sbox S_5 from [6], is

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\sigma[x]$	0	A	6	5	8	D	3	4	7	C	2	E	9	F	B	1

For this Sbox, we get the following results (the corresponding transition matrices are given in [7]).

– For $r_1 = r_2 = 2$, i.e., for six rounds, the highest probability is obtained for four differentials, namely when $(\Delta_{in}, \Delta_{out}) \in \{(5,2),(2,5)\} \times \{(5,2),(2,5)\}$. The corresponding probability is 2^{-50}.
– For $r_1 = 2$ and $r_2 = 3$, i.e., for seven rounds, the probability of the best differential is 2^{-58}.

4 Key recovery on round reduced versions of PRINCE

In this section we show how the previously described differentials can be extended by up to 4 full rounds, in order to recover the key of reduced variants of PRINCE, in a chosen-plaintext scenario.

4.1 General principle

Our attack applies to r rounds of PRINCE with $r > 4$. We call *reduced cipher*, the cipher derived from r rounds of PRINCE$_{core}$ by removing, both at the beginning and at the end of the cipher, one full round and an additional nonlinear layer. Therefore, the reduced cipher corresponds to the $(r-4)$-round function defined in the previous section $\mathcal{F}^{K_1}_{r_1+r_2+2}$ with $r_1 + r_2 + 2 = r - 4$ and $|r_1 - r_2| \leq 1$. Our attack exploits together several differentials with the same input and output activity patterns for the reduced cipher, as proposed in [2, 3]. Indeed, there exist several differentials with the same square activity pattern which have similar probabilities. In particular, if $(\delta_1, \delta_2) \mapsto (\delta'_1, \delta'_2)$ holds with probability p, so do $(\delta_2, \delta_1) \mapsto (\delta'_1, \delta'_2)$, $(\delta_1, \delta_2) \mapsto (\delta'_2, \delta'_1)$ and $(\delta_2, \delta_1) \mapsto (\delta'_2, \delta'_1)$.

These square differentials for $(r - 4)$ rounds of the reduced cipher can be exploited to perform a key-recovery attack on r rounds of PRINCE. Actually, due to the very simple key schedule of PRINCE, the knowledge of 66 key bits only (out of 128) allows to determine whether a given plaintext-ciphertext pair corresponds to a pair of input and output of the reduced cipher which follows one of the considered differentials. Moreover, there is an efficient procedure for deriving, from each plaintext-ciphertext pair, the list of all partial key candidates which lead to one of the expected differentials for the reduced cipher. Assuming the considered differentials have good probabilities, the correct partial key is then suggested with higher probability than the other ones. Our attack then follows the general principle of statistical attacks on block ciphers: the first part is a distillation phase which counts how many times the partial keys are suggested by the available plaintext-ciphertext pairs. Then the analysis phase selects the partial keys suggested by at least τ pairs, where the threshold τ is a parameter of the attack. The search phase eventually consists in finding the key of the cipher from the identified list of partial keys.

For the sake of clarity, we first focus on input and output differences corresponding to the $[0,0]$-activity pattern. From now on, we denote by $\varDelta = (\varDelta_{in}, \varDelta_{out})$ a differential with input and output having the $[0,0]$-activity pattern, and by $p_{true}(\varDelta)$ its transition probability. We consider a set $\varSigma$ of D such differentials. $\varSigma_{in}$ denotes the set of all input differences $\varDelta_{in}$ such that $(\varDelta_{in}, \varDelta_{out})$ belongs to $\varSigma$ for some $\varDelta_{out}$. Similarly, $\varSigma_{out}$ is the set of all output differences in $\varSigma$. The sizes of these sets are denoted by D_{in} and D_{out} respectively.

4.2 Extension of the reduced cipher square differentials

Construction of structures. Our attack makes use of structures of plaintexts. A structure S_{P^1,P^3} is a set of 2^{32} pairs of plaintexts and ciphertexts, defined as follows. Columns 1 and 3 of all the plaintexts in the structure share the same values P^1 and P^3, while Columns 0 and 2 take all the $(2^{16})^2 = 2^{32}$ possible values. Building such a structure requires 2^{32} encryption queries.

Let us consider any differential $\varDelta = (\varDelta_{in}, \varDelta_{out})$ in $\varSigma$. For any plaintext P in the structure, we denote by X the (unknown) value obtained during the encryption of P after addition of the whitening key W_{in}, and by U the (unknown) value at the beginning of the reduced cipher. Similarly, with any ciphertext C we associate the value Y before the addition of the whitening key W_{out} and the value V at the end of the reduced cipher (see Figure 4).

Then, the plaintext $P_\varDelta$ corresponding to $U \oplus \varDelta_{in}$ has the same value as P on Columns 1 and 3. Indeed these columns do not depend on the active nibbles of $\varDelta_{in}$ when computing backwards as shown on Figure 4. Therefore, $P_\varDelta$ lies in the same structure as P. The subset of all $2^{31}(2^{32}-1)$ pairs of distinct plaintexts obtained from S_{P^1,P^3} can be partitioned into subsets of 2^{31} pairs, each corresponding to a given difference at the beginning of the reduced cipher. As a consequence, there are exactly 2^{31} pairs of distinct plaintexts from S_{P^1,P^3} which lead to any given input difference $\varDelta_{in} \in \varSigma_{in}$ for the reduced cipher. We can build N_s such structures for a cost of $N = 2^{32}N_s$ chosen plaintexts.

Right pairs and candidate pairs. When considering the target key, the pairs that follow one of the differentials in $\varSigma$ are called *right pairs*. Their number is about

$$N_{true} = 2^{31} N_s \sum_{\varDelta \in \varSigma} p_{true}(\varDelta) = \frac{N}{2} p_{true} \text{ where } p_{true} = \sum_{\varDelta \in \varSigma} p_{true}(\varDelta) , \qquad (1)$$

i.e., p_{true} is the sum of the probabilities of all the differentials in $\varSigma$.

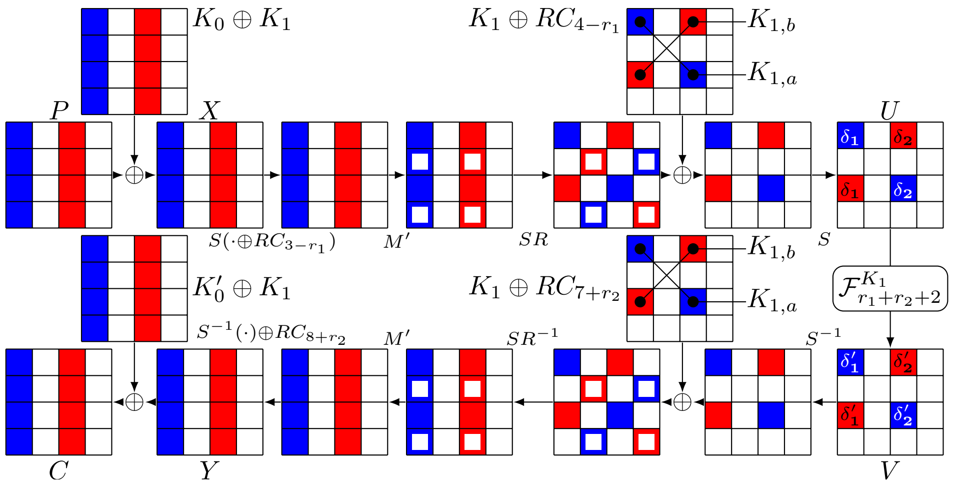

Fig. 4. Extension of the reduced cipher square differentials. The blue (resp. red) nibbles correspond to plaintext/ciphertext column 0 (resp. 2). The colored nibbles of the states can be recovered from the corresponding colored nibbles of the key material, plaintext and ciphertext. In the cipher states, full colored nibbles indicate active nibbles, emptied colored nibbles and empty nibbles indicate no difference.

For the right pairs, the difference at the end of the reduced cipher has the $[0, 0]$-activity pattern. Then, it can be seen on Figure 4 that this difference can propagate only to Columns 0 and 2 of the ciphertexts. Thus, our criterion to detect potential right pairs is a collision on Columns 1 and 3 of the ciphertexts. All pairs with ciphertexts colliding on Columns 1 and 3 are named *candidate pairs*. Obviously, all right pairs belong to the set of candidate pairs. However, about $2^{63}N_s$ possible pairs can be obtained from the N_s structures, and each of those fulfills the criterion with probability 2^{-32}. Therefore we have about $F = 2^{31}N_s$ candidate pairs.

Guessing key bits. For any given candidate pair (P, P', C, C'), it can be easily seen that the difference $U \oplus U'$ at the beginning of the reduced cipher can be recovered from the plaintexts and from Columns 0 and 2 of the pre-whitening key $W_{in} = K_0 \oplus K_1$ and from four nibbles of K_1 (the colored nibbles in Figure 4), namely the bits $K_1^{[20;23]\cup[28;31]\cup[52;55]\cup[60;63]}$. Similarly, the difference at the end of the reduced cipher depends on the ciphertexts, on the same four nibbles of K_1 and also on Columns 0 and 2 of the post-whitening key $W_{out} = K'_0 \oplus K_1$.

Since the two whitening keys are related by some simple formula deduced from the fact that $K'_0 = (K_0 \ggg 1) \oplus (K_0 \gg 63)$, the following property can be easily proved by induction.

Property 1. Let j and ℓ be two indices with $0 \leq j \leq \ell \leq 63$. From the knowledge of K_1^{ℓ} (or $K_0^{\ell+1}$, with $K_0^{64} = K_0^0$) and $W_{in}^{[j;\ell]}$ and $W_{out}^{[j;\ell]}$, one can derive linearly the values of $K_0^{[j;\ell]}$ and $K_1^{[j;\ell]}$ (with a perturbation by K_0^{63} at bit position 0).

Using Property 1, it is easy to establish that the key material required to ensure a candidate pair follows a differential corresponds to 66 information bits of the key: bits $K_1^{[16;31]\cup[48;63]}$ and bits $K_0^{0\cup[16;32]\cup[48;63]}$.

Distillation phase overview. Taking into account the structure used by the attack, the distillation phase basic principle is given in Algorithm 1.

Algorithm 1 Distillation phase of the attack.

```
for all $N_s$ structures of plaintexts
    for all pairs $(P, P')$ in the structure such that $(C^1 \oplus C'^1 = C^3 \oplus C'^3) = 0$
        for all 66-bit partial keys $k$ which lead to $(U \oplus U', V \oplus V') \in \Sigma$
            Increment the counter $N_k$
```

We show in the following subsections how the partial keys corresponding to candidate pairs can be identified efficiently during the distillation phase. We remark that a second distillation phase can be performed by considering another input-output activity pattern for the differentials of the reduced cipher, and we study the distribution of the counters N_k as a function of the data complexity and a threshold parameter τ.

4.3 Identification of key candidates in the distillation phase

Overview. The attack described above can be performed only if, for each candidate pair, the partial keys which lead to a differential in Σ for the reduced cipher can be efficiently determined. Finding these key candidates is not a trivial task in the general case. In the following, we describe a method to achieve it with a time complexity close to D encryptions per candidate pair, where D is the number of differentials considered in the attack.

First, let us remark that the first (resp. last) two rounds of the cipher, before the beginning (resp. the end) of the reduced cipher $\mathcal{F}_{r_1+r_2+2}^{K_1}$, act independently on the different columns (or diagonals) of the state, as depicted on Figure 4. Thus, for every differential $\Delta \in \Sigma$, for column $i \in \{0, 2\}$ and for every partial values of K_1 ($K_{1,a}$ or $K_{1,b}$), one can precompute all compatible input-output column pairs $(X^i, X^i \oplus \Delta X^i, Y^i, Y^i \oplus \Delta Y^i)$ for the cipher without its pre- and post-whitening, as detailed in the precomputation step paragraph below.

Then, the processing step successively examines all candidate pairs of plaintext-ciphertext for all considered differentials of the reduced cipher. Since the whitening keys satisfy $W_{in} = P \oplus X$ and $W_{out} = C \oplus Y$, the precomputed tables enable us to determine all partial values of the whitening keys which can lead to a differential in Σ for the reduced cipher. Among these whitening keys, further filtering is required to ensure constraints stemming from the key schedule are satisfied.

Now, we describe in details how lists can be precomputed and carefully merged in order to reduce the complexity of the attack. The list elements are tuples of values. We will assume that the lists are sorted lexicographically after being generated.

Precomputation step. At the input of the reduced cipher, the four active nibbles can be divided into two pairs. Indeed, referring to Figure 4, column X^0 (resp. X^2) of X can be computed from the two blue (resp. red) active nibbles of U, the two blue (resp. red) nibbles of K_1 and two additional inactive nibbles in the blue (resp. red) column after the M' layer. Therefore, we precompute the following lists: for all $\Delta_{in} = (\delta_1, \delta_2) \in \Sigma_{in}$, $\mathcal{L}^{in,a}_{\Delta_{in}}$ is composed of triples $(K_{1,a}, \Delta X^0, X^0)$ where X^0 and ΔX^0 are two 16-bit elements and $K_{1,a}$ is an 8-bit part of K_1 corresponding to its two blue nibbles on Figure 4. The values X^0 and $(X^0 \oplus \Delta X^0)$ are the values on Column 0 of some internal states X and X' which lead to Δ_{in} on the blue nibbles at the input of the reduced cipher given $K_{1,a}$. Similarly, $\mathcal{L}^{in,b}_{\Delta_{in}}$ contains triples $(K_{1,b}, \Delta X^2, X^2)$ where $K_{1,b}$ corresponds to the two red nibbles of K_1, and X^2 and $(X^2 \oplus \Delta X^2)$ are the values on Column 2 of some X and X' which lead to Δ_{in} on the red nibbles at the input of the reduced cipher. The detailed algorithm for computing $\mathcal{L}^{in,a}_{\Delta_{in}}$ is given in Algorithm 2.

Algorithm 2 Precomputation step of the distillation phase

```
$\mathcal{L} \leftarrow \{(z_0, z_2, z'_0, z'_2) \in (\mathbf{F}_2^4)^4 \ : \sigma(z_0) \oplus \sigma(z'_0) = \delta_1$ and $\sigma(z_2) \oplus \sigma(z'_2) = \delta_2$ with $(\delta_1, \delta_2) = \Delta_{in}\}$
for all $(z, z')$ in $\mathcal{L}$
  for all $K_{1,a} = (k_{0,0}, k_{2,2})$ corresponding to the two blue nibbles of $K_1$
    for all pairs of nibbles $(z_1, z_3)$
      $X^0 \leftarrow S^{-1}(M'(z_0 \oplus k_{0,0} \oplus RC^{0,0}_{4-r_1}, z_1, z_2 \oplus k_{2,2} \oplus RC^{2,2}_{4-r_1}, z_3)) \oplus RC^{0}_{3-r_1}$
      $\Delta X^0 \leftarrow X_0 \oplus S^{-1}(M'(z'_0 \oplus k_{0,0} \oplus RC^{0,0}_{4-r_1}, z_1, z'_2 \oplus k_{2,2} \oplus RC^{2,2}_{4-r_1}, z_3)) \oplus RC^{0}_{3-r_1}$
      Add $(K_{1,a}, \Delta X^0, X^0)$ to $\mathcal{L}^{in,a}_{\Delta_{in}}$
```

The first list $\mathcal{L}$ has size exactly 2^8. Then, each list $\mathcal{L}^{in,a}_{\Delta_{in}}$ contains 2^{24} elements, as well as the lists $\mathcal{L}^{in,b}_{\Delta_{in}}$. Similarly, we compute the $2D_{out}$ lists $\mathcal{L}^{out,a}_{\Delta_{out}}$ and $\mathcal{L}^{out,b}_{\Delta_{out}}$, each of size 2^{24} elements, which correspond to the possible pairs of values for Y and Y' on Column 0 and Column 2, respectively.

For a PRINCE computation, the eight bits of K_1 involved in $\mathcal{L}^{out,a}_{\Delta_{out}}$ (resp. in $\mathcal{L}^{out,b}_{\Delta_{out}}$) are the same as the ones involved in $\mathcal{L}^{in,a}_{\Delta_{in}}$ (resp. in $\mathcal{L}^{in,b}_{\Delta_{in}}$). Therefore, the lists sharing the same key bits can be merged. For any $\Delta = (\Delta_{in}, \Delta_{out})$, the two lists $\mathcal{L}^{in,a}_{\Delta_{in}}$ and $\mathcal{L}^{out,a}_{\Delta_{out}}$ then lead to a list $\mathcal{L}^{a}_{\Delta}$ of size 2^{40} composed of tuples $(\Delta X^0, \Delta Y^0, T, K_{1,a}, X^0, Y^0)$. T is a 3-bit linear tag appended to each element of the list, whose role will be explained later. The overall time complexity of this merging process is proportional to $2^{40}D$. The memory required for storing the D lists $\mathcal{L}^a_\Delta$ corresponds to $2^{40}D$ elements of 75 bits. Similarly, the lists $\mathcal{L}^{in,b}_{\Delta_{in}}$ and $\mathcal{L}^{out,b}_{\Delta_{out}}$ are merged into D lists $\mathcal{L}^b_\Delta$ of size 2^{40}.

Processing step. We now explain how, for each candidate pair, (P, P', C, C'), we compute all partial keys which lead to a differential in Σ for the reduced cipher. Since all possible values for Columns 0 and 2 of (X, X', Y, Y') have been precomputed and $P \oplus P' = X \oplus X', C \oplus C' = Y \oplus Y'$, the precomputed lists provide us with the list of all possible values for the whitening keys $W_{in} = K_0 \oplus K_1$ and $W_{out} = K'_0 \oplus K_1$ on Columns 0 and 2, along with co-determined values of some bits of Columns 0 and 2 of K_1.

Moreover, since for all $i \neq 0$ we have $X_i \oplus P_i = K^i_0 \oplus K^i_1$ and $Y_i \oplus C_i = K^{i+1}_0 \oplus K^i_1$, we also deduce the following relation between P, C, X, Y and K_1:

$$P_i \oplus C_{i-1} = X_i \oplus Y_{i-1} \oplus K^i_1 \oplus K^{i-1}_1 \ . \tag{2}$$

For each element in one of the lists $\mathcal{L}^a_\Delta$, the right-hand term in the previous relation is known for $61 \leq i \leq 63$ since the nibble in Position $(0, 0)$ of K_1 belongs to $K_{1,a}$. This is the 3-bit value, denoted by T, included in the tuples in $\mathcal{L}^a_\Delta$. Then, for each candidate pair examined in the attack, the corresponding value of $(P_i \oplus C_{i-1})$, $61 \leq i \leq 63$ is compared with T. The number of candidates for the value of W^0_{in} is then divided by a factor 8, on average. Similarly, the 3-bit value corresponding to the right-hand side of (2) for $28 \leq i \leq 31$ is included in each element of $\mathcal{L}^b_\Delta$. The 66-bit keys corresponding to a given candidate pair are then computed through Algorithm 3. The principle of this algorithm is to generate lists $\mathcal{L}^i$, $i \in \{0, 2\}$ containing partial key values affecting mainly Column i and compatible with Column i of the considered candidate pair, to compute the key bits shared by elements in both lists, and finally to merge the two lists according to the shared bit values.

For $\Delta \in \Sigma$, the number of elements in lists $\mathcal{L}^a_\Delta$ and $\mathcal{L}^b_\Delta$, which take a given value on the first 3 values ($\Delta X, \Delta Y$ and T) of their tuple, amounting to 35 bits, is 2^5 on average. For each of these elements, we need to compute 4 bits by a simple linear relation. The final merging step between $\mathcal{L}^0$ and $\mathcal{L}^2$ leads to 2^2 key candidates on average. It requires to sort the two lists according to their first two values, which can be done in linear time, for a cost of 2^6.

Thus the average complexity of Algorithm 3 is $2^6 D$ elementary operations and the average number of partial key candidates found for each candidate pair is $2^{-8} \times (2^5)^2 D = 4D$.

4.4 Iterating the distillation phase for recovering the whole key

We have shown how to perform a distillation phase leveraging differentials with input and output $[0, 0]$-activity patterns. It extracts information about the number of values taken by 66 key bits from structures of plaintext-ciphertext pairs.

Algorithm 3 Processing step of the distillation phase

```
**Input**: a candidate pair $(P, P', C, C')$
$B \leftarrow (P_i \oplus C_{i-1}, 61 \leq i \leq 63)$; $B' \leftarrow (P_i \oplus C_{i-1}, 29 \leq i \leq 31)$.
**for all** $\Delta \in \Sigma$
  $\mathcal{L}^0 = \emptyset, \mathcal{L}^2 = \emptyset$
  **for all** elements in $\mathcal{L}^a_\Delta$ of the form $(P^0 \oplus P'^0, C^0 \oplus C'^0, B, K_{1,a}, X^0, Y^0)$
    $W^0_{in} \leftarrow X^0 \oplus P^0$; $W^0_{out} \leftarrow Y^0 \oplus C^0$.
    Compute $K_1^{[52;55]}$ with Property 1 starting from $K_1^{60} \in K_{1,a}$.
    Add $(K_1^{[52;55]}, K_1^{[20;23]}, W^0_{in}, W^0_{out}, K_1^{63})$ to $\mathcal{L}^0$ ($K_1^{[20;23]}$ is a nibble of $K_{1,a}$).
  **for all** elements in $\mathcal{L}^b_\Delta$ of the form $(P^2 \oplus P'^2, C^2 \oplus C'^2, B', K_{1,b}, X^2, Y^2)$
    $W^2_{in} \leftarrow X^2 \oplus P^2$; $W^2_{out} \leftarrow Y^2 \oplus C^2$.
    Compute $K_1^{[20;23]}$ with Property 1 starting from $K_1^{28} \in K_{1,b}$.
    Add $(K_1^{[52;55]}, K_1^{[20;23]}, W^2_{in}, W^2_{out}, K_1^{31})$ to $\mathcal{L}^2$ ($K_1^{[52;55]}$ is a nibble of $K_{1,b}$).
  **for all** pairs of elements in $\mathcal{L}^0, \mathcal{L}^2$ matching on their first two entries
    Increment the counter $N_k$, with $k = (W^0_{in}, W^2_{in}, W^0_{out}, W^2_{out}, K_1^{63}, K_1^{31})$
```

We now remark that a second distillation phase can be performed, by adapting the procedure described in previous subsections to the case of $[0,1]$-activity pattern instead of the $[0,0]$-activity pattern. Then, the bits of the key involved in the computation of the active nibbles cover mainly Columns 1 and 3 (instead of 0 and 2).

In this second distillation phase, the conformity of a plaintext-ciphertext pair to one of the differentials in Σ can also be checked by guessing key nibbles and whitening keys columns. They can be shown to be equivalent to 66 bits of key information: bits $K_1^{[1;15]\cup[32;47]}$, $K_0^{[1;16]\cup[32;48]}$, $K_0^0 \oplus K_0^{63}$, and $K_1^0 \oplus K_0^{63}$.

Then, each of the 128 key bits is involved in at least one of the distillation phases. Each distillation phase restricts the value taken by about half of the key.

4.5 Complexity analysis

Distillation phase complexity. For reasons of symmetry, the number of differentials D that we take into account and the corresponding probabilities p_{true} are the same in both parts. We also take in both cases the same value for the number of structures N_s. In each distillation step, Algorithm 3 is applied once for each candidate pair. Since the time complexity of the distillation phases is assessed in elementary operations (xor on 4 bits), the comparison with the complexity of the generic attack is not easy. We argue that the search for all partial key candidates for each candidate pair and each differential, which has been estimated to be 2^6 operations, is less costly than one full encryption. Indeed, each of the 11 applications of the linear layer M' required by a single encryption consists of four linear transformations on 16 bits. This obviously corresponds to more than 2^8 binary xors. Then, the time complexity of the distillation phase satisfies

$$\mathsf{Time}_1 = 2 \times 2^{31} N_s \times 2^6 D \text{ elementary operations } \leq \frac{2N_{true}}{p_{true}} D \text{ encryptions.}$$

Since each distillation phase requires specific structures, we have

$$\mathsf{Data} = 2 \times 2^{32} N_s = \frac{4N_{true}}{p_{true}} \; .$$

The memory complexity of our attack comes essentially from the storage of partial key candidates counters during the distillation phase. It is worth noticing that the total number of partial key candidates considered during the distillation is bounded by $4D \times 2^{31} N_s$. Using an appropriate data structure to access the counters, the memory required is then bounded by

$$\mathsf{Memory} \leq 4D \times \frac{N_{true}}{p_{true}} \; .$$

Search phase complexity. We now want to determine to what extent the statistic on the keys resulting from the distillation phases reduces the size of the search space.

All counters N_k correspond to the sum of some basic counters $N_k(P, P', C, C')$, one for each candidate pair. These basic counters are defined as

$$N_k(P, P', C, C') = \begin{cases} 1 & \text{if } (U \oplus U', V \oplus V') \in \Sigma \\ 0 & \text{otherwise.} \end{cases}$$

The basic counters for wrong keys follow a Bernoulli distribution. Indeed, let us consider any wrong key k. For each candidate pair, columns P^1, P'^1, P^3 and P'^3 are chosen according to the structure and $C^1 = C'^1$ and $C^3 = C'^3$. The values of columns 0 and 2 of P, P', C and C' are distributed uniformly among pairs that fulfill $P \neq P'$ and $C \neq C'$. The values of active diagonals at the beginning and at the end of the reduced cipher when encrypting P, P' and decrypting C, C' under key k follow the same distribution. The counter N_k is incremented if one of the D differences in Σ occurs at the beginning and at the end of the reduced cipher, which happens with probability $(2^{32} - 1)^{-2} D \approx 2^{-64} D = p_{false}$.

For the right key guess, the average value of N_k is N_{true}. Let us choose a threshold $\tau = N_{true}/\eta$. If τ decreases, the success probability of the attack increases, but so does the number of wrong keys reaching the threshold. Theorem 3 of [2] gives an estimation of the probability that a wrong key reaches this threshold, using accurate bounds for the tail distribution of the counter values for wrong keys. Using our notation, this estimation leads to $\Pr[N_k \geq \tau] \approx G(p_{true}/\eta D_{in}, p_{false}/D_{in}, \tau)$, where

$$G(x, y, t) = e^{\frac{-tD(x||y)}{x}} \left[\frac{x(1-y)}{(x-y)\sqrt{2\pi t(1-x)}} + \frac{1}{\sqrt{8\pi t}} \right] ,$$

and $D(x||y) = x \log\left(\frac{x}{y}\right) + (1-x) \log\left(\frac{1-x}{1-y}\right)$ is the Kullback-Leibler divergence.

Taking into account that $p_{true} \ll 1$ and $p_{false} \ll 1$, and denoting by ρ the ratio p_{false}/p_{true}, it can be shown that

$$\Pr[N_k \geq N_{true}] \approx (\eta\rho)^{\tau} e^{\tau(1-\eta\rho)} \frac{1}{\sqrt{2\pi\tau}} \left(\frac{1}{2} + \frac{1}{1-\eta\rho} \right) \approx \frac{3(e \cdot \eta\rho)^{\tau}}{\sqrt{8\pi\tau}} \; ,$$

as in most cases, $p_{false} \ll p_{true}/\eta$.

After the two distillation phases, the whole key is eventually recovered by a search phase which only examines the keys having their two 66-bit restrictions above the threshold. These keys can be easily found by merging the two lists of partial keys which share the same value on the following 4 bits: $K_0^{48}, K_0^{32}, K_0^{16}$ and $(K_0^0 \oplus K_0^{63})$.

The complexity, Time_2, of the search phase corresponds to one encryption under each key whose score reaches the threshold in both distillation phases. There are approximately 2×2^{62} wrong keys that collide with the right key on all the 66 key bits involved in one of the distillation phases, therefore

$$\mathsf{Time}_2 \approx 2^{128}\left(\frac{3(e\cdot\eta\rho)^{\tau}}{\sqrt{8\pi\tau}}\right)^2 + 2^{63}\frac{3(e\cdot\eta\rho)^{\tau}}{\sqrt{8\pi\tau}} \approx 2^{125}\frac{9(e\cdot\eta\rho)^{2\tau}}{\pi\tau}.$$

Overall complexity. Taking into account the complexity of distillation and search phases, we get

$$\mathsf{Data} \times (\mathsf{Time}_1 + \mathsf{Time}_2) = \frac{8N_{true}^2 D}{p_{true}^2} + \frac{9\times 2^{127} N_{true}}{\pi\cdot p_{true}\cdot\tau}\left(\frac{e\cdot p_{false}\cdot N_{true}}{p_{true}\cdot\tau}\right)^{2\tau}.$$

Memory. The distillation phase is the only phase having a large memory requirement, thus the memory complexity is

$$\mathsf{Memory} = 4D\frac{N_{true}}{p_{true}} = D\cdot\mathsf{Data}\ .$$

4.6 Success probability of the attack

We now estimate the success probability of our attack. For each distillation phase, there are $t = 2^{63}p_{true}$ right pairs, that might follow one of the differentials in Σ. Provided t does not exceed 2^{16}, these pairs all belong to different structures with a high probability. Therefore, t is a good estimation of the number of structures (out of the 2^{32} possible choices) that contain at least one valid pair for each of the distillation phases.

During the attack, the adversary chooses N_s structures out of 2^{32} in both phases, and succeeds if each set of N_s structures contains at least τ right pairs. The probability that a set of N_s structures contains exactly u right pairs can be approximated by the probability that it contains exactly u structures chosen among those containing a right pair. By summing for $u \geq \tau$, and assuming independence of the two phases, this leads to the following estimation of the probability of success:

$$P(N_s,\tau) = \left(\sum_{u\geq\tau}\frac{\binom{t}{u}\binom{2^{32}-t}{N_s-u}}{\binom{2^{32}}{N_s}}\right)^2 = \left(\sum_{u\geq\tau}\frac{\binom{N_s}{u}\binom{2^{32}-N_s}{t-u}}{\binom{2^{32}}{t}}\right)^2\ . \qquad (3)$$

5 Experimental estimation of p_{true}

In this section we display some experimental results that validate our attack strategy. Indeed, the complexity evaluation of our attack heavily relies on the estimation of the value of p_{true}. However, this estimation highly relies on the assumption that the differential transitions between the rounds are independent. In the case of PRINCE, as the round keys are all identical, this assumption is questionable. Therefore we have validated our approach by the following experiments.

In the next section, we show that, based on theoretical estimations of p_{true} and of the time and memory complexities, the best choice of parameters for attacking 10 rounds of PRINCE are $N_{true} = 6$ and $D = 12$. This attack then makes use of 24 differentials over 6 rounds, 12 in each distillation phase. Each of these differentials have been determined by aggregating several differential paths with a high probability over 6 rounds.

We have implemented an algorithm that computes exhaustively the pairs of messages which follow one of these differential paths, given one value of the key. As a result, we obtain a lower bound on the number of pairs that follow one of the 12 differentials for each distillation phase.

We ran our program on 1000 randomly chosen keys. The results are depicted on Figure 5, where the x-coordinate (resp. the y-coordinate) represents the number of pairs following one of the differentials involved in the first (resp. the second) distillation phase. Following our theoretical estimation, the average number of pairs should be $\sum 2^{63} p_{true}(\Delta) = p_{true} 2^{63} = 800$ for each phase. In our experiments we observe an average value of 869 pairs for the first phase and 861 pairs for the second phase. This tends to show that the differentials we have identified are followed with probabilities slightly higher than estimated.

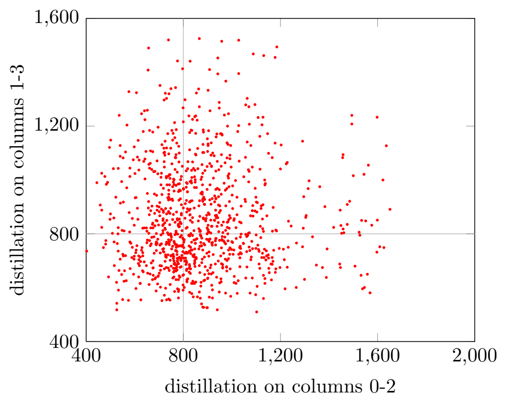

Fig. 5. Number of message pairs following one of the 16 differentials for both distillation phases for the 1000 keys tested

6 Results

In this section we apply the previously described attack to several variants of PRINCE. Using the study of the differential properties of PRINCE displayed in Section 3, we still have to select some parameters for our attack, namely D (the number of differentials we take into account), τ (the threshold score for key candidates), and N_{true} (the estimated score of the right keys). By choosing $\tau = N_{true}$, the right key reaches τ in each distillation phase with probability close to 0.5, leading to an almost constant success probability of 0.25.

Then, our goal is to find the values of D and N_{true} that minimize the product $\mathsf{Comp} = \mathsf{Data} \times (\mathsf{Time}_1 + \mathsf{Time}_2)$.

The following table summarizes our most significant results. We focus on 9-round and 10-round versions of PRINCE with the original Sbox. We also emphasize that some members of the PRINCE-family are more vulnerable by presenting results obtained with the modified Sbox defined in Section 3. It is worth noticing that, in all cases, p_{true} is much higher than the false alarm probability $p_{false} = 2^{-64}D$.

Cipher	rounds	D	p_{true}	N_{true}	Data	Time_1	Time_2	Comp
PRINCE original Sbox	9	40	$2^{-43.30}$	3	$2^{46.89}$	$2^{51.21}$	$2^{41.34}$	$2^{98.10}$
PRINCE original Sbox	10	12	$2^{-53.36}$	6	$2^{57.94}$	$2^{60.53}$	$2^{56.54}$	$2^{118.56}$
PRINCE modified Sbox	10	12	$2^{-46.83}$	3	$2^{50.42}$	2^{53}	$2^{52.08}$	$2^{104.03}$
PRINCE modified Sbox	11	12	$2^{-54.81}$	8	$2^{59.81}$	$2^{62.40}$	$2^{56.92}$	$2^{122.24}$

Our choice of parameters has been optimized for $\tau = N_{true}$. If we increase the amount of data, keeping the same threshold value would highly increase the complexity of the search phase. A better strategy may then consist in increasing the threshold but keeping it less than N_{true}. In this case, we increase the success probability of the attack. For example, with the previous parameters, if the amount of available data is $\mathsf{Data} = 2^{59.7}$ (instead of $2^{57.94}$), N_{true} increases from 6 to 20. Then, choosing $\tau = 7$ leads to an overall complexity of $\mathsf{Comp} = 2^{120}$, and a success probability of 0.85 as given by Equation (3).

References

1. Farzaneh Abed, Eik List, and Stefan Lucks. On the Security of the Core of PRINCE Against Biclique and Differential Cryptanalysis. IACR Cryptology ePrint Archive, Report 2012/712, 2012. `http://eprint.iacr.org/2012/712`.
2. Céline Blondeau and Benoît Gérard. Multiple differential cryptanalysis: Theory and practice. In *FSE 2011*, volume 6733 of *LNCS*, pages 35–54. Springer, 2011.
3. Céline Blondeau, Benoît Gérard, and Kaisa Nyberg. Multiple Differential Cryptanalysis Using LLR and χ^2 Statistics. In *SCN 2012*, volume 7485 of *LNCS*, pages 343–360. Springer, 2012.
4. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES 2007*, volume 4727 of *LNCS*. Springer, 2007.

5. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications. In *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 208–225. Springer, 2012.
6. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - a low-latency block cipher for pervasive computing applications (full version). IACR Cryptology ePrint Archive, Report 2012/529, 2012. `http://eprint.iacr.org/2012/529`.
7. Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. Multiple differential cryptanalysis of round-reduced PRINCE (Full version). IACR Cryptology ePrint Archive, Report 2014/089, 2014. `http://eprint.iacr.org/2014/089`.
8. Anne Canteaut, María Naya-Plasencia, and Bastien Vayssière. Sieve-in-the-Middle: Improved MITM techniques. In *CRYPTO 2013 (I)*, volume 8042 of *LNCS*, pages 222–240. Springer, 2013.
9. Pierre-Alain Fouque, Antoine Joux, and Chrysanthi Mavromati. Multi-user collisions: Applications to Discrete Logs, Even-Mansour and Prince. IACR Cryptology ePrint Archive, Report 2013/761, 2013. `http://eprint.iacr.org/2013/761`.
10. Henri Gilbert and Thomas Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In *FSE 2010*, volume 6147 of *LNCS*, pages 365–383. Springer, 2010.
11. Zheng Gong, Svetla Nikova, and Yee Wei Law. KLEIN: A New Family of Lightweight Block Ciphers. In *RFIDSec*, volume 7055 of *LNCS*, pages 1–18. Springer, 2011.
12. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J.B. Robshaw. The LED Block Cipher. In *CHES 2011*, volume 6917 of *LNCS*, pages 326–341. Springer, 2011.
13. Jérémy Jean, Ivica Nikolic, Thomas Peyrin, Lei Wang, and Shuang Wu. Security Analysis of PRINCE. In *FSE 2013*, LNCS. Springer, 2013. To appear.
14. Leibo Li, Keting Jia, and Xiaoyun Wang. Improved meet-in-the-middle attacks on AES-192 and PRINCE. IACR Cryptology ePrint Archive, Report 2013/573, 2013.
15. Marine Minier and Henri Gilbert. Stochastic cryptanalysis of Crypton. In *FSE 2000*, volume 1978 of *LNCS*, pages 121–133. Springer, 2000.
16. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher CLEFIA. In *FSE 2007*, volume 4593 of *LNCS*, pages 181–195. Springer, 2007.
17. Hadi Soleimany, Céline Blondeau, Xiaoli Yu, Wenling Wu, Kaisa Nyberg, Huiling Zhang, Lei Zhang, and Yanfeng Wang. Reflection Cryptanalysis of PRINCE-like Ciphers. In *FSE 2013*, LNCS. Springer, 2013. To appear.
18. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE : A Lightweight Block Cipher for Multiple Platforms. In *SAC 2012*, volume 7707 of *LNCS*, pages 339–354. Springer, 2012.
19. Wenling Wu and Lei Zhang. LBlock: A Lightweight Block Cipher. In *ACNS 2011*, volume 6715 of *LNCS*, pages 327–344. Springer, 2011.