THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 16, Special Issue 2015, pp. 253-264

NEW APPROACHES FOR ROUND-REDUCED PRINCE CIPHER CRYPTANALYSIS

Raluca POSTEUCA[1], Cristina-Loredana DUTA[2], Gabriel NEGARA[3]

[1]University of Bucharest, Romania
[2]University Politehnica of Bucharest, Romania
[3]Al. I. Cuza University of Iasi, Romania
E-mail: cristina.duta.mapn@outlook.com

The rapid development of technology influences the future of pervasive computing, which is characterized by the usage of smart devices with limited resources, such as computing power, memory or battery supply. To create pervasive applications, which meet real-time security needs, ciphers that are capable to run on devices with small computing power must be created. In this context, in the last few years, lightweight cryptography has been a hot research area and many impressive implementations and ciphers have been published. At ASIACRYPT'2012 conference a new lightweight block cipher, called PRINCE, was presented. After its appearance, the designers of the algorithm proposed a challenge for all cryptographers, which consist in finding practical attacks on round-reduced PRINCE cipher. In this article, we present various new attacks on round-reduced PRINCE and the time and data complexity necessary for each of them. The first attack which was successfully applied to 4, 5 and 6-rounds is the boomerang attack. We also describe a known-plaintext attack for 4 and 5-round PRINCE algorithm.

Key words: lightweight cryptography, PRINCE cipher, boomerang attack, known-plaintext attack.

1. INTRODUCTION

Lightweight cryptography is an emerging domain of symmetric cryptography, which rapidly became popular due to the need of secure constrained devices. For instance, RFID tags on commercial products, network sensors, traffic jam detectors, cardiac pacemakers all have in common the necessity for a secure and privacy friendly mode of operation, under an environment with restricted resources. All the previously existing cryptographic primitives, such as hash functions, classical stream and block ciphers are not able to fulfil the requirements of devices with limited resources (low latency, low power usage and low hardware implementation). In this context, new innovative algorithms with unconventional designs, called lightweight ciphers, were designed to ensure security for tight constraints. The main characteristics of these new primitives should be: reduced power consumption (in many situations we deal with limited battery), sufficient speed (provide communication in real time) and small footprints (low construction complexity and low gate number).

Because lightweight cryptographic primitives have to be as small as possible and also ensure low power usage, requirements that imply a trade-off between lightweightness and security are necessary (reach high level of security using only a small computing power). Due to their characteristics, it is extremely important to understand and analyse the security of lightweight ciphers.

The security of a lightweight block cipher can be determined by analysing the results obtained for practical and theoretical attacks such as meet-in-the middle attacks, differential attacks, integral attacks and so on. Nowadays, the need of secure cryptographic algorithms that can be implemented in devices with limited resources has made the industry more focused on practical attacks. Compared with the industry, the academia is mainly interested in theoretical attacks which offer important information about the security ensured by block ciphers, about their components and about their design.

Some of the most known and best studied lightweight stream ciphers are Mickey [1], Grain [2] and Trivium [3]. The lightweight block cipher category includes primitives such as CLEFIA [4], PRESENT [5], Hight [6], KATAN [7], LED [8], PRINTCipher [9], Klein [10], Piccolo [11] and KTAN-TAN [7]. Hash

functions for lightweight cryptography were also recently developed: QARK [12], SPONGENT [13], and PHOTON [14]. A more recent lightweight block cipher, published in 2012 is PRINCE [15], which has a unique property called α-reflection (the decryption process is reduced to an encryption with a slightly different key). Moreover, PRINCE algorithm is considered to be the first block cipher that has as main priority latency for hardware implementations. The algorithm is briefly described in the following section.

Our contribution. We have focused our research in studying and implementing various types of attacks on round-reduced PRINCE that have different data and time complexities. We present new boomerang attacks on 4, 5 and 6 round PRINCE with a time complexity of maximum 2^{34} encryptions and a relatively low data complexity. Our boomerang attacks use differential characteristics based on which we are able to build boomerang distinguishers which can easily be exploited. Moreover, we were able to improve the key recovery phase by exploiting several properties of the substitution layer of PRINCE algorithm. An important aspect of our research is the fact that all these attacks are confirmed by various experimental results. Furthermore, while significant progress has been done in previous PRINCE cryptanalysis, as far as we know, we are the first to successfully design a classical boomerang attack on round-reduced PRINCE block cipher.

We also designed and implemented known-plaintext attacks for 4 and 5 rounds. The disadvantage of the attack for 5 round PRINCE cipher is the fact that the data complexity is very high, which is why the attack can only be considered theoretical.

Organization of the paper. The paper is organized as follows. The PRINCE cipher is briefly described in Section 2. In Section 3 we present the related work regarding theoretical and practical attacks on round-reduced or full round PRINCE algorithm. Section 4 includes the details of our boomerang attacks and the results we obtained. The known-plaintext attacks are described in Section 5. In Section 6 we present a comparison between our implementations results and others researchers' results and we evaluate the complexity of the attacks. Finally, Section 7 contains our conclusions and future work.

2. DESCRIPTION OF PRINCE

PRINCE [15] is a lightweight block cipher introduced in 2012 at ASIACRYPT International Conference on the Theory and Application of Cryptology and Information Security. The proposed cipher has 64-bit blocks, uses a key *k* of 128 bits and is based on the so-called *FX* construction [16].

2.1. Key Schedule

The master key *k* is divided into two 64-bits keys, k_0 and k_1, such that $k = (k_0 \| k_1)$, where || denotes the concatenation. *k* is extended to 192-bits according to the following equation, where L represents a simple linear transformation.

$$k = (k_0 \| k_1) \rightarrow (k_0 \| k_0' \| k_1) = (k_0 \| L(k_0) \| k_1); L(k_0) = (k_0 >>> 1) \oplus (k_0 >> 63) \quad (1)$$

The third generated subkey k_0', together with k_0 are used as whitening keys. The key k_1 of 64-bits is used as an internal key for the core of the block cipher (12 rounds) referred to as PRINCE$_{core}$. The general structure of PRINCE block cipher is depicted in Fig. 1.

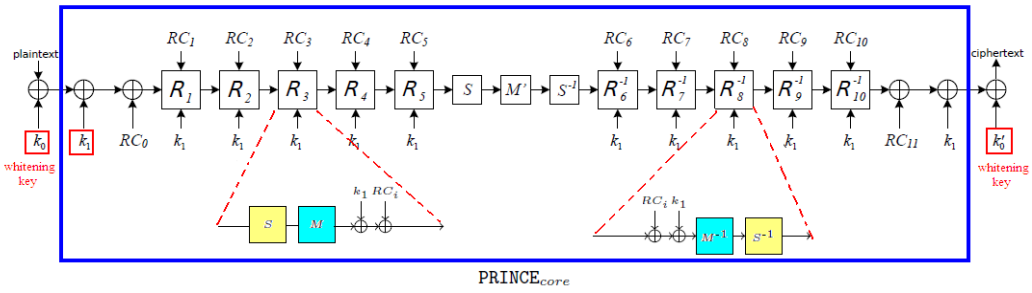

Fig. 1 – Structure of PRINCE block cipher.

2.2. PRINCE$_{core}$

The PRINCE$_{core}$ is actually a Substitution-Permutation-Network which has 12 rounds. The 64-bit state can be seen as a 4x4 array of nibbles, but in our implementations the state is seen as a 16-nibble array. Each round function, R_i, includes: a S-box layer (denoted S), a linear layer (denoted M), a key addition operation and addition of a round constant.

The **substitution layer S** applies a 4-bit S-box S to every nibble of the internal state. The S-box is shown in Table 1 in hexadecimal notation.

Table 1

S-box of PRINCE block cipher

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

The **linear layer M** includes, as described in equation 2, the multiplication by a 64x64 matrix M' and a *Shift Rows* (SR) transformation (similar to AES's).

$$M = SR \circ M'. \tag{2}$$

In equation 3 is explained how the multiplication with the matrix M' is performed, where M_0 and M_1 are two different matrixes of 16x16 that have the following property: after the multiplication, each output bit depends only on three bits of the input value. $(x_0 \| x_1 \| x_2 \| x_3)$ represents the 64-bit state.

$$M' \cdot (x_0 \| x_1 \| x_2 \| x_3) = M_0 \cdot (x_0) \| M_1 \cdot (x_1) \| M_1 \cdot (x_2) \| M_0 \cdot (x_3) \tag{3}$$

SR permutes the 16 nibbles according to Table 2.

Table 2

SR permutation of PRINCE block cipher

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
								↓							
0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

Key addition represents the bitwise addition between the current state $b_{63}...b_0$ and the 64-bit round key $k_1 = k_{63}...k_0$. The operation performed is defined in equation 4.

$$b_j \to b_j \oplus k_j, 0 \le j \le 63 \tag{4}$$

Round constant addition can be expressed as in equation 5. It is the XOR operation between the 64-bit state $b_{63}...b_0$ with the 64-bit round constant $RC_i = \upsilon_{63}^i...\upsilon_0^i, 0 \le i \le 11$.

$$b_j \to b_j \oplus \upsilon_j^i, 0 \le j \le 63 \tag{5}$$

2.3. PRINCE$_{core}$ encryption and decryption processes

The ***encryption*** process includes three types of rounds: forward rounds, middle rounds and backward rounds. The equations for each type of round are described below.

$$Forward: R_i(x) = M(S(x)) \oplus k_1 \oplus RC_i, i = 0...4 \tag{6}$$

$$Middle\ \ 2\ \ rounds: R(x) = S^{-1}(M' \cdot S(x)) \tag{7}$$

$$Backward: R_i^{-1}(x) = S^{-1}(M^{-1}(x \oplus k_1 \oplus RC_i)), i = 7...11 \tag{8}$$

The ***decryption*** can be performed by reusing the encryption process with a different key. This functionality is successful due to the α-reflection property of PRINCE block cipher. This means that the round constants satisfy equation 9, where α is the constant hexadecimal value *0xc0ac29b7c97c50dd*.

$$RC_i \oplus RC_{11-i} = \alpha, 0 \le i \le 11 \tag{9}$$

Moreover, this allows that in hardware and software implementations, the same encryption and decryption operations to be applied. For the expanded key $(k_0 \| k_0^{'} \| k_1)$ the relation between the encryption and decryption process can be written as in equation 10.

$$D_{(k_0 \| k_0^{'} \| k_1)}(\cdot) = E_{(k_0 \| k_0^{'} \| k_1 \oplus \alpha)}(\cdot) \tag{10}$$

3. RELATED WORK

With the emergence of all these new lightweight ciphers, new attacks were also designed. This section includes the results published by various researchers regarding theoretical and practical attacks on round-reduced or full round PRINCE lightweight block cipher.

In [17], the authors describe several approaches to determine the algorithm's security against the following: related-key attack on full round PRINCE, related-key boomerang attack on PRINCE$_{core}$, single-key boomerang attack on PRINCE$_{core}$ for a chosen α and integral attacks for 4, 5 and 6 round PRINCE algorithm and PRINCE$_{core}$. The related-key boomerang attack has been applied to full round PRINCE$_{core}$, with data and time complexity of 2^{39}. This attack was conducted because at that moment, the designers of PRINCE algorithm didn't make any claims regarding related-key attack model. Moreover, they applied single-key boomerang attack for a modified value of α, combining the related-key boomerang attack with the usage of the α-reflection property of the core block cipher. They were able to recover the 128-bit master key using 2^{32} memory, 2^{63} operations and 2^{33} data. As a conclusion for [17], it is specified that these practical attacks are successful for PRINCE$_{core}$ and if they are applied to PRINCE, the time complexity will have an exponential increase up to 2^{64}.

The authors in [18] concentrated their efforts on applying meet-in-the-middle (MitM) attacks on round-reduced PRINCE, on developing an algebraic attack based on a SAT solver and on developing differential attack which has a reduced data complexity. The MitM attacks were successfully conducted for 6, 8 and 10 rounds PRINCE cipher. While the attacks on 6 and 8 rounds have data and time complexity relatively small, for 10 rounds the data complexity is 2^{57} and it requires 2^{68} encryptions, but it has a memory complexity of 2^{41} (measured in 64-bit blocks). For the last mentioned attack, the authors obtained a data complexity of $2^{14.9}$, a time complexity of $2^{32.9}$ and a memory complexity less than 2^{27} for 6 round PRINCE cipher.

In [19] several MitM attacks on AES-192 and PRINCE cipher are described. The authors introduce a new technique which uses a 6-round distinguisher to attack 8-round PRINCE and PRINCE$_{core}$. The previously mentioned attack on 8-round PRINCE was successful and required 2^{53} chosen plaintexts, 2^{28} 64-bit blocks of memory and 2^{53} encryption operations. For 9-round PRINCE, a 7-round distinguisher was used and for its success were necessary 2^{57} chosen plaintexts, $2^{57.3}$ memory blocks and 2^{64} encryptions. In [20] a reflection cryptanalysis is presented, which demonstrated that the security of PRINCE cipher and other algorithms similar to it is highly dependent on the value of α. The authors were able to perform a key-recovery attack for full round PRINCE with time complexity of $2^{72.39}$ and data complexity of $2^{57.98}$, using for this attack a weak class of α. In [21] is presented one of the best known attacks, in terms of the number of rounds. A 10-round attack is described with time complexity of $2^{60.62}$ and data complexity of $2^{57.94}$, which exploits some properties of the algorithm and uses multiple differentials. Moreover, the authors have found an S-box that allows the attack to be extended up to 11 rounds, with a small increase (almost 2^2) of the data and time complexity. In [22] the authors present a sieve-in-the-middle algorithm that is used to attack 8-round PRINCE cipher and which needs for its success only 2 known plaintext-ciphertext pairs.

In [23], the authors present a differential cryptanalysis attack for 6-round PRINCE$_{core}$. Also, a biclique attack is performed for full round PRINCE$_{core}$ cipher which has a small memory complexity, of only 2^8 64-bit blocks and data complexity of 2^{40}. A 4-round differential attack is also presented, but this has high data (2^{48}) and time complexity ($2^{56.26}$) and also requires a large memory space of 2^{48}.

A differential fault attack is described in [24]. This attack exploits the properties of the linear layer and can determine the entire 128-bit key using an average of 7 fault injections.

The attacks we have developed (3 new boomerang attacks and 2 known-plaintext attacks on round-reduced PRINCE) will be described further on.

4. BOOMERANG ATTACK

The boomerang attack [25] is an adaptive chosen plaintext attack which is particularly effective when there exists a high-probability differentials for both first half of encryption and first half of decryption. Unlike simple differential cryptanalysis where we use only a high probability differential trail (we exploit how the input differences can affect the outputted ciphertexts), the boomerang attack uses differentials that can be used to cover only a part of the cipher. For the attack to be successful, we need *quartets* (P, P', Q and Q') with some particular properties. In order to obtain this type of *quartets* for a boomerang attack the following steps must be performed:

1. Choose a random plaintext denoted P and calculate P' as $P' = P \oplus \Delta$.
2. Perform the encryptions of P and P' from which we obtain $C = E(P)$ and $C' = E(P')$.
3. Calculate $D = C \oplus \nabla$ and $D' = C' \oplus \nabla$.
4. Perform the decryptions of D and D' from which we obtain $Q = E^{-1}(D)$ and $Q' = E^{-1}(D')$.
5. At the end, Q and Q' are compared. If the differentials hold, then the relation $Q \oplus Q' = \Delta$ is true.

4.1. The 4-round PRINCE cipher attack

The first step in our boomerang attack on 4-round PRINCE was to identify two high probability differentials, which are presented in Fig. 2.

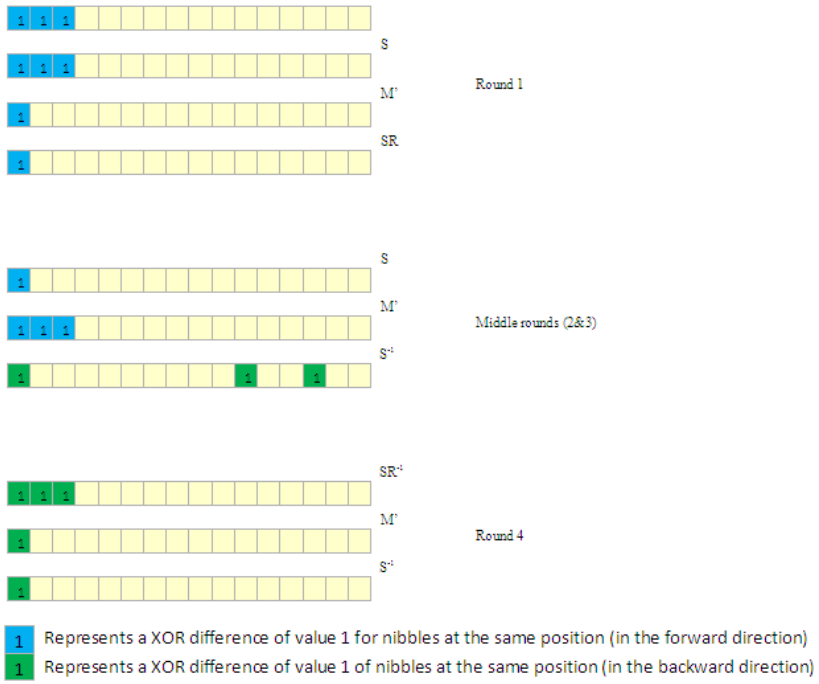

Fig. 2 – The two differential trails used for 4-round PRINCE boomerang attack.

The identified forward differential trail has a probability of 2^{-8} and the backward differential trail has a probability of 2^{-2}. In this context, the probability of finding a right quartet is of approximately 2^{-12}.

For this scenario, the forward input difference, Δ, has the first three nibbles active and the backward input difference, ∇, has the first nibble active. Moreover, all the active nibbles have a difference of value 1. The pseudocode of the attack is presented further on in Fig. 3.

I. **for** i between 0 and *total_number_of_plaintexts*
   1. Randomly *select* a plaintext P
   2. *Compute* P' as $P' = P \oplus \Delta$
   3. *Encrypt* P as $C = E(P)$ using 4-round PRINCE algorithm
   4. *Encrypt* P' as $C' = E(P')$ using 4-round PRINCE algorithm
   5. *Compute* D as $D = C \oplus \nabla$
   6. *Compute* D' as $D' = C' \oplus \nabla$
   7. *Decrypt* D as $Q = E^{-1}(D)$
   8. *Decrypt* D' as $Q' = E^{-1}(D')$
   9. *Compute* $Q \oplus Q' = \Delta'$
   10. **if** $\Delta' = \Delta$ **then** store the quartet (P, P', Q, Q')

II. **foreach** *quartet* stored in the *quartet_list*
   1. Compute the corresponding ciphertexts quartet (C, C', D, D')
   2. **foreach** value (from 0 to 15) of the first nibble of the key $k_0 \oplus k_1$
   3. partially decrypt, through the S layer, the first nibble of each ciphertext of the quartet (C, C', D, D')
   4. After this decryption we compare: the first nibble of C with the first nibble of D
      the first nibble of C' with the first nibble of D'
   5. **if** $C_{first\_nibble} \oplus D_{first\_nibble} = 1$ and $C'_{first\_nibble} \oplus D'_{first\_nibble} = 1$ **then** we have found a possible correct nibble of the key and we increment a counter for its frequency

III. Based on frequencies' values, we select as correct, the value of the nibble with the highest frequency.

Fig. 3 – Boomerang attack pseudocode for 4-round PRINCE cipher.

Regarding the attack's implementation, the following observations are important.

a. We never obtained a single value with the maximum frequency. In fact, we obtained each time four values. In order to improve these results, we have slightly modified the backward differential trial, by setting the value of the first nibble of ∇ to be 2 instead of 1. This modified value remains local, it does not propagate through the trail. With this change, the probability of the backward trail will be 2^{-3} and the probability of finding a right quartet will become approximately 2^{-15}. Using this new trail, after applying the attack on 4-round PRINCE, we obtained only two possible correct values for the first nibble of the key;
b. To avoid obtaining many false positives in our implementation, the *total_number_of_plaintexts* had the value 2^{18};
c. For this attack, we have obtained a reasonable data complexity of 2^{18} and a low time complexity of approximately 2^{20}.

4.2. The 5-round PRINCE cipher attack

Based on the previous presented attack, we have extended the number of rounds of PRINCE cipher from 4 to 5, by adding a round before the middle rounds. In this scenario, we used the high probability differential trails which are presented in Fig. 4.

The identified forward differential trail has the probability 2^{-14} and the backward differential trail has the probability 2^{-2}. In this context, the probability of finding a right quartet is of approximately 2^{-18}.

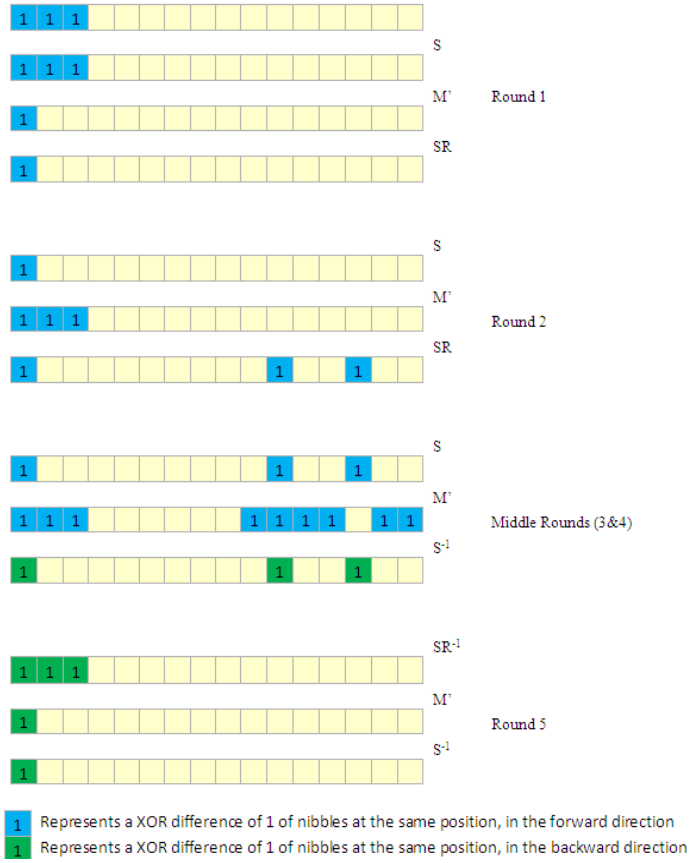

Fig. 4 – The two differential trails used for 5-round PRINCE boomerang attack.

For this scenario, the forward input difference Δ has the first three nibbles active and the backward input difference ∇ has the first nibble active. Moreover, all the active nibbles have a difference of value 1.

The pseudocode of this version of the boomerang attack is similar with the previous one, with some small modifications such as: we run the encryption process for 5-round PRINCE algorithm, instead of 4-round PRINCE and the *total_number_of_plaintexts* has the value 2^{25} instead of 2^{18}.

Compared with the 4-round attack, the data complexity is multiplied by a factor of 2^7 and the time complexity is also increased by a factor of 2^7.

4.3. The 6-round PRINCE cipher attack

The previous boomerang attack for 5-round PRINCE cipher can be extended to a 6-round PRINCE attack by adding one final round to the encryption process, ensuring in this way that we have two forward rounds, two middle rounds and two backward rounds. The high probability differential trails which we used for this attack are shown in Fig. 5.

The forward differential trail has a probability of 2^{-14} and the backward differential trail has a probability of 2^{-8}. In this context, the probability of finding a right quartet is of approximately 2^{-30}.

For this scenario, both the forward input difference Δ and the backward input difference ∇ have the first three nibbles active. Moreover, all the active nibbles have a difference of value 1.

The pseudocode of this version of the boomerang attack is similar with the one presented in Fig. 3, with some differences. For instance, the encryption process has 6 rounds, instead of 4 rounds and the *total_number_of_plaintexts* has the value 2^{32} instead of 2^{18}.

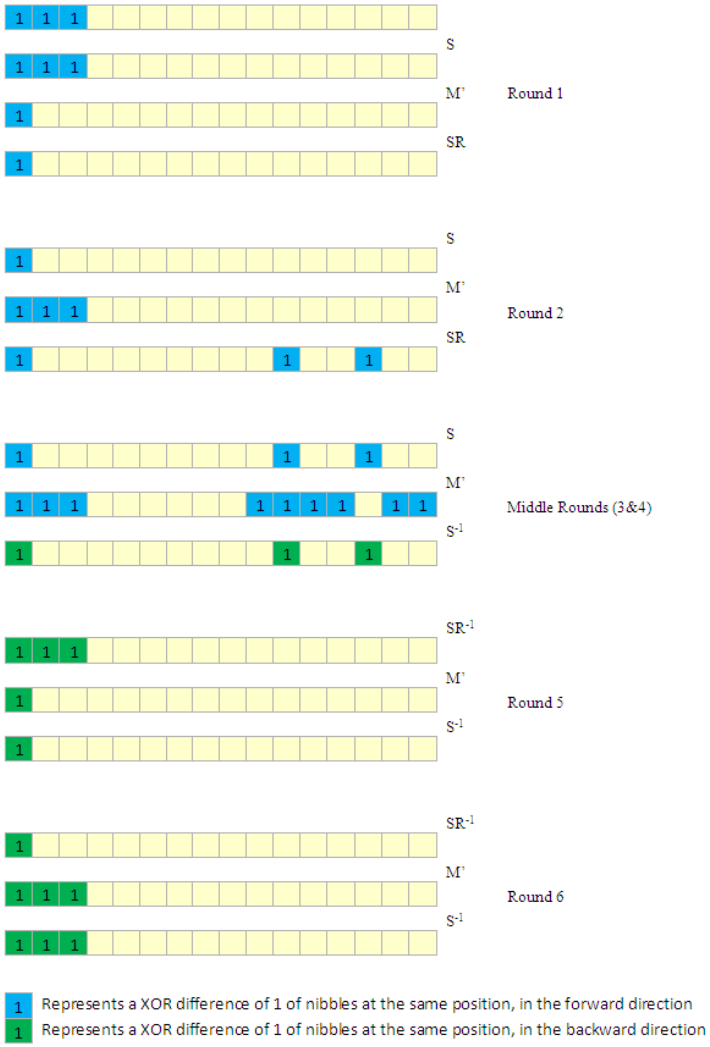

Fig. 5 – The two differentials trails used for 6-round PRINCE boomerang attack.

The data complexity increases by a factor of 2^7 compared with 5-round boomerang attack and the time complexity reaches 2^{34}.

5. KNOWN-PLAINTEXT ATTACK

We have designed a guess-and-determine attack on 4-rounds PRINCE which is based on the fact that the same key is used in every round of the encryption process (we conduct an exhaustive search on the first half of the key) and we also exploited some algebraic properties of the M' layer - the equations (11) and (12).

5.1. The 4-round PRINCE cipher attack

The attack was initially designed for 4-rounds PRINCE$_{core}$, as presented in Fig. 6:

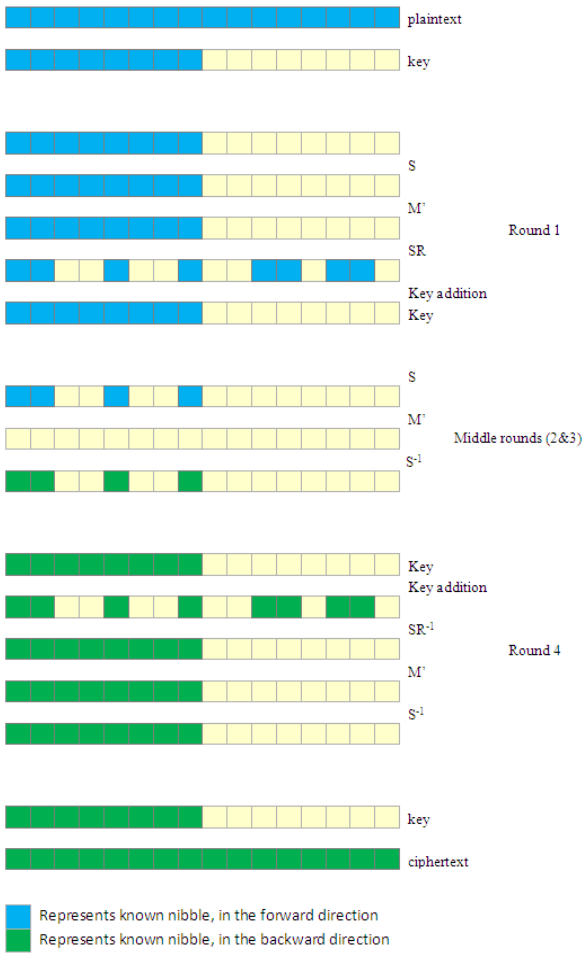

Fig. 6 – The two differentials trails used for 4-round PRINCE known-plaintext attack.

The pseudocode of the attack is presented further on.

I. **for** i between 0 and *total_number_of_plaintexts*
   6. Randomly *select* a plaintext P
   7. *Encrypt* P as $C = E(P)$ using 4-round PRINCE algorithm
   8. **foreach** value (from 0 to 15) of the first 8 nibbles of the key (denoted *partial_key*)
      i. partially encrypt P, through a round and through a S layer
      ii. partially decrypt C, through a round and through a S layer
      *at this stage, we have obtained the input and output data of the M' layer
      iii. **if** equations (11) and (12) are verified, **then** we increment a frequency counter for the current *partial_key*

II. **foreach** *partial_key* in *partial_key_list*
   i. **if** *partial_key* counter is equal to $total\_number\_of\_plaintexts$ **then** the *partial_key* is probable right key

Fig. 7 – Known-plaintext attack pseudocode for 4-round PRINCE cipher.

We denoted the array input for the M' layer as in_data and the array output for the M' layer as *out_data*. Based on these notations, the following equations can be written and hold for every correct *out_data=M'·in_data*.

$$(in\_data[0] \oplus in\_data[1]) \& 4 = (out\_data[0] \oplus out\_data[1]) \& 4 \quad (11)$$

$$(in\_data[4] \oplus in\_data[7]) \& 8 = (out\_data[4] \oplus out\_data[7]) \& 8 \quad (12)$$

In order to avoid obtaining too many false positives, we set the *total_number_of_plaintexts* to 2^5 in our implementation. In this context, we obtained a single value with the maximum frequency.

For this attack, we have obtained a low data complexity of 2^5 and a time complexity of 2^{37}.

At this point, based on our known-plaintext attack we are able to recover 32 bits of the key k_1. The remaining 32 bits can be obtained using the same plaintext-ciphertext pairs to perform an exhaustive search for the last 8 nibbles of the key. Moreover the total time complexity will become 2^{38} (increasing with only a factor of 2) while the data complexity will remain the same.

Even though we only took into consideration the PRINCE$_{core}$, this attack can be mounted for full 4-round PRINCE cipher. For this extended attack, instead of performing an exhaustive search for 32 bits of k_1, we search for the first 32 bits of $k_1 \oplus k_0$ and 32 bits of k_1 (in order to compute the first 32 bits of k_0) and the last bit of k_0 (to compute the first 32 bits of k_0'). At this point, we are able to recover 65 bits from $(k_0 \| k_1)$ and if we apply a similar method with the one previously described (for recovering k_1 for PRINCE$_{core}$), we will determine the entire $(k_0 \| k_1)$ key. The total time complexity is approximately 2^{66} and we expect that the data complexity will also increase (there will be more than 2^5 plaintexts necessary to obtain a single value with the maximum frequency).

5.2. The 5-round PRINCE cipher attack

The 4-round PRINCE known-plaintext attack previously presented can be extended for 5-round PRINCE, by adding one initial round. The difference between this attack and the attack described for 4-round PRINCE is that the exhaustive search will be performed for 64 bits of $k_1 \oplus k_0$ (based on this, we can fully encrypt the first round, and the attack can be reduced to the attack presented in Fig. 7) and 33 bits of k_1. Because the time complexity will increase exponentially, up to $2^{97} \cdot$ *total_number_of_plaintexts*, we consider this to be a theoretical attack.

6. SUMMARY OF RESULTS

Our results are summarized in Table 3 together with the results of some major previous results of similar types of attacks on round-reduced PRINCE lightweight block cipher.

Table 3

Summary of attacks on round-reduced PRINCE and PRINCE$_{core}$

GENERAL DESCRIPTION				COMPLEXITY	
Reference	Cipher	Rounds	Attack type	Time	Data
Our attack (Section 5)	PRINCE	4	Known-Plaintext	2^{66}	$> 2^5$
		5		$> 2^{97}$	$> 2^5$
P. Derbez		4		2^{43}	2^5
Our attack (Section 5)	PRINCEcore	4		2^{38}	2^5
Our attack (Section 4)	PRINCE	4	Boomerang	2^{20}	2^{18}
		5		2^{27}	2^{25}
		6		2^{34}	2^{32}

Observations:
We measure the memory complexity in 64-bit blocks
We measure time complexity in encryption operations

In comparison with the known-plaintext attack designed by Derbez, our attack on 4-round requires more than 2^5 plaintexts and has a time complexity higher than 2^{43} (by a factor of 2^{23}). Regarding boomerang attacks, there are some security analysis made for PRINCE block cipher, but these concentrate on particular types of boomerang attacks such as related-key boomerang and single-key boomerang for chosen α. In this context, our paper introduces the first single-key boomerang attack on round-reduced PRINCE (4, 5 and 6 rounds). Data and time complexity of our attacks are small enough for them to be considered practical attacks (both are smaller than 2^{34}).

7. CONCLUSIONS

In this article we propose some new attacks for round-reduced PRINCE lightweight block cipher. In particular, we have successfully designed and implemented boomerang and known-plaintext attacks. We have applied the boomerang attack on 4, 5 and 6 rounds and the known-plaintext attack on 4 and 5 rounds and we compared the data and time complexities we have obtained with publicly known results. To the best of our knowledge these are the first published results for single-key boomerang attacks on round-reduced PRINCE.

Our future work will include searching some new approaches to reduce the data and time complexity of the previously described attacks and also extending these attacks on a higher number of rounds (if possible, to full round PRINCE).

ACKNOWLEDGEMENTS

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/187/1.5/S/155536.

REFERENCES

1. S. BABBAGE, M. DODD, *The MICKEY Stream Ciphers*, New Stream Cipher Designs – The eSTREAM Finalists, **4986**, 2008.
2. M. HELL, T. JOHNASSEN, W. MEIER, *GRAIN: A Stream Cipher for Constrained Environments*, International Journal of Wireless and Mobile Computing, **2**, pp. 86-93, 2007.
3. C. CANNIERE, B. PRENEEL, *Trivium Specifications*, eSTREAM ECRYPT Stream Cipher Project, 2006.
4. T. SHIRAI, K. SHIBUTANI, T. AKISHITA, S. MORIAI, T. IWATA, *The 128-bit Block Cipher CLEFIA (Extended Abstract)*, Fast Software Encryption - FSE, **4593**, pp. 181-195, 2007.
5. A. BOGDANOV, L. R. KNUDSEN, G. LEANDER, C. PAAR, A. POSCHMANN, M. J. B. ROBSHAW, Y. SEURIN, C. VIKKELSO, *PRESENT: An Ultra-Lightweight Block Cipher*, Cryptographic Hardware and Embedded Systems – CHESS, **4727**, pp. 450-466, 2007.
6. C. CANNIERE, O. DUNKELMAN, M. KNEZEVIC, *KATAN and KTANTAN-A – A Family of Small and Efficient Hardware-Oriented Block Ciphers*, Cryptographic Hardware and Embedded Systems – CHESS 2009, **5747**, pp. 4272-288, 2009.
7. D. HONG, J. SUNG, S. HONG, J. LIM, S. LEE, B. KOO, C. LEE, D. CHANG, J. LEE, K. JEONG, H. KIM, J. KIM, S. CHEE, *HIGHT: A New Block Cipher Suitable for Low-Resource Devices*, Cryptographic Hardware and Embedded Systems – CHESS, **4249**, pp. 46-59, 2006.
8. J. GUO, T. PEYRIN, A. POSCHMANN, M. J. B. ROBSHAW, *The LED Block Cipher*, Cryptographic Hardware and Embedded Systems – CHESS, **6917**, pp. 326-341, 2011.
9. L. KNUDSEN, G. LEANDER, A. POSCHMANN, M. J. B. ROBSHAW, *PRINTCipher: A Block Cipher for IC-Printing*, Cryptographic Hardware and Embedded Systems – CHESS, **6225**, pp. 16-32, 2010.
10. Z. GONG, S. NIKOVA, Y. W. LAW, *KLEIN: A New Family of Lightweight Block Ciphers*, RFID Security and Privacy – RFIDSec 2011, **7055**, pp. 1-18, 2011.
11. K. SHIBUTANI, T. ISOBE, H. HIWATARI, A. MITSUDA, T. AKISHITA, T. SHIRAI, *Piccolo: An Ultra-Lightweight Block Cipher*, Cryptographic Hardware and Embedded Systems – CHESS, **6917**, pp. 342-357, 2011.
12. J. P. AUMASSON, *QUARK: A Lightweight Hash*, Journal of Cryptology, **26**, pp. 313-339, 2013.
13. A. BOGDANOV, M. KNEZEVIC, G. LEANDER, D. TOZ, K. VARICI, I. VERBAUWHEDE, *SPONGENT: A Lightweight Hash Function*, Cryptographic Hardware and Embedded Systems –CHESS, **6917**, pp. 312-325, 2011.
14. J. GUO, T. PEYRIN, A. POSCHMANN, *The PHOTON Family of Lightweight Hash Functions*, Crypto, **6841**, pp. 222-239, 2011.

15. J. BORGHOFF, A. CANTEAUT, T. GUNEYSU, E. B. KAVUN, M. KNEZEVIC, L. R. KNUDSEN, G. LEANDER, V. NIKOV, C. PAAR, C. RECHBERGER, P. ROMBOUTS, S. S. THOMSEN, T. YALCIN, *PRINCE – A Low-latency Block Cipher for Pervasive Computing Applications*, Advances in Cryptology – ASIACRYPT, pp. 208-225, 2012.
16. A. BIRYUKOV, *DES-X (or DESX)*, Encyclopedia of Cryptography and Security (2nd Ed.), pp. 331, 2011.
17. J. JEAN, I. NIKOLIC, T. PEYRIN, L. WANG, S. WU, *Security Analysis of PRINCE*, Proceedings of Fast Software Encryption, pp. 92-111, 2013.
18. P. DERBEZ, L. PERRIN, *Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE*, Proceedings of Fast Software Encryption 2015.
19. L. LI, K. JIA, X. WANG, *Improved Meet-in-the-Middle Attacks on AES-192 and PRINCE*, Cryptology ePrint Archive, Report 2013/573, 2013.
20. H. SOLEIMANY, C. BLONDEAU, X. YU, W. WU, K. NYBERG, H. ZHANG, L. ZHANG, Y. WANG, *Reflection Cryptanalysis of PRINCE-like Ciphers*, Proceedings of Fast Software Encryption, pp. 71-91, 2013.
21. A. CANTEAUT, T. FUHR, H. GILBERT, M. NAYA-PLASENCIA, J. R. REINHARD, *Multiple differential cryptanalysis of round-reduced PRINCE*, Cryptology ePrint Archive, Report 2014/089, 2014.
22. A. CANTEAUT, M. NAYA-PLASENCIA, B. VAYSSIERE, *Sieve-in-the-Middle: Improved MITM Attacks (Full Version)*, Proceedings of CRYPTO 2013, pp. 222-240, 2013.
23. A. FARZANEH, L. EIK, L. STEFAN, *On the security of the core of Prince against biclique and differential cryptanalysis*, Cryptology ePrint Archive, Report 2012/712, 2012.
24. L. SONG, L. HU, *Differential Fault Attack on the PRINCE Block Cipher*, Cryptology ePrint Archive, Report 2013/043, 2013.
25. D. WAGNER, *The boomerang attack*, Proceedings of Fast Software Encryption - FSE, pp. 156-170, 1999.