Information Processing Letters 112 (2012) 624–629

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock ☆

Marine Minier [a,*], María Naya-Plasencia [b]

[a] Université de Lyon, INRIA, CITI, F-69621, France
[b] University of Versailles, France

ARTICLE INFO

Article history:
Received 8 February 2012
Received in revised form 18 April 2012
Accepted 23 April 2012
Available online 15 May 2012
Communicated by D. Pointcheval

Keywords:
Analysis of algorithms
Cryptography
Attack
Block cipher
Differential cryptanalysis

ABSTRACT

LBlock is a new lightweight block cipher proposed by Wu and Zhang (2011) [12] at ACNS 2011. It is based on a modified 32-round Feistel structure. It uses keys of length 80 bits and message blocks of length 64 bits.

In this letter, we examine the security arguments given in the original article and we show that we can improve the impossible differential attack given in the original article on 20 rounds by constructing a 22-round related key impossible differential attack that relies on intrinsic weaknesses of the key schedule. This attack has a complexity of 2^{70} cipher operations using 2^{47} plaintexts. This result was already published in Minier and Naya-Plasencia (2011) [9].

1. Introduction

During the last five years, many lightweight block ciphers for constrained environments have been proposed. We could cite: PRESENT [1], HIGH [6], DESL [7], CGEN [10], KATAN & KTANTAN [3], SEA [11], LED [5], KLEIN [4] and LBlock [12].

Even if some cryptanalytic results (see [2,8] for example) have already appeared concerning the security of those particular block ciphers, it still remains necessary to intensively study their security and their efficiency. Moreover, when designing lightweight block ciphers, the design of the key schedule must be carefully studied. The reason is that it is not always possible to store the round-keys generated by the key schedule on small platforms due to their limited memory. In that case, the round-keys must be generated "on the fly". This problem has been carefully addressed in the case of CGEN [10].

Table 1
Comparison table of cryptanitic results against LBlock.

Attack	Nb rounds	In	Time complexity
Imp. Diff.	20	[12]	$2^{72.7}$
Integral	20	[12]	$2^{63.7}$
RK Imp. Diff.	22	This paper	2^{70}

In this paper, we focus on the security evaluation of the new lightweight block cipher LBlock [12]. We show how the original impossible differential attack proposed in the LBlock article can be extended by two rounds (up to 22 rounds in all) using a related key impossible differential attack. Of course, an attack in the related key security model is much weaker than an attack in the secret key setting but this result provides a better understanding of the LBlock security evaluation, namely concerning its key schedule. (See Table 1.)

This paper is organized as follows: Section 2 gives a brief description of the LBlock lightweight block cipher, Section 3 describes the related key impossible differential attack on 22 rounds of LBlock. Finally, Section 4 concludes this paper.

☆ This work was partially supported by the French National Agency of Research: ANR-11-INS-011.

* Corresponding author.
E-mail addresses: marine.minier@insa-lyon.fr (M. Minier), maria.naya.plasencia@gmail.com (M. Naya-Plasencia).

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2012.04.012

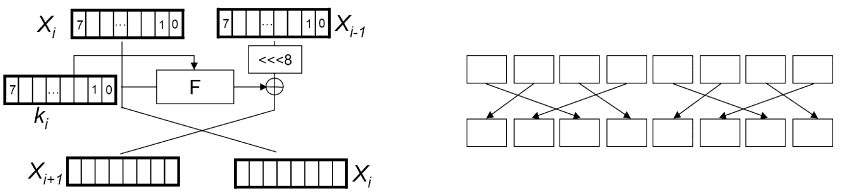

Fig. 1. On the left: Overview of one modified Feistel round of LBlock. On the right, the permutation P seen at nibble level.

2. Description of LBlock

LBlock is a new lightweight block cipher presented by Wu and Zhang at ACNS 2011 [12]. It uses 80-bit keys and 64-bit blocks and is based on a modified 32-round Feistel structure (see Fig. 1).

The round function F first computes $X_i \oplus k_i$ and then applies a transformation S (composed of 8 parallel applications of 8 different 4-bit bijective S-boxes) and a permutation P (that exchanges the places of the nibbles as shown on Fig. 1).

The key schedule takes as input a master key K seen as a key register denoted at bit level as $K = K_{79}K_{78}\cdots K_0$ and outputs 32 round subkeys k_i. It repeats the following steps for $i = 1$ to 31 knowing that k_1 is initialized with the 32 leftmost bits of the key register K:

1. $K \lll 29$.
2. $[K_{79}K_{78}K_{77}K_{76}] = S_9[K_{79}K_{78}K_{77}K_{76}]$ where S_9 is the ninth S-box.
3. $[K_{75}K_{74}K_{73}K_{72}] = S_8[K_{75}K_{74}K_{73}K_{72}]$ where S_8 is the eighth S-box.
4. $[K_{50}K_{49}K_{48}K_{47}] = [K_{50}K_{49}K_{48}K_{47}] \oplus [i]_2$
5. k_{i+1} is selected as the leftmost 32 bits of the key register K.

3. Related key impossible differential attacks on 22 rounds of LBlock

The attack described in this section is a related key impossible differential attack. This analysis takes advantage of a 15-round impossible differential path and of some weaknesses of the key schedule. In this section, we first introduce the used related key differential sets and the impossible differential path. We then provide the complete description of the attack.

3.1. Related key sets

The details of the related key sets are given in Appendix A. The main properties of those related keys come from some intrinsic properties of the key schedule. First, when a low weight difference is introduced in a pair of keys, those differences do not cross the S-boxes every round but in average only every 9 rounds (among 32). Moreover, an injected difference will appear in average only every three subkeys, creating low weight differential paths. Thus, we are able to construct related keys differential paths with a very low general weight (the ones presented in Appendix A have only between 12 and 15 active nibbles on all the 32 subkeys). In summary, the diffusion is not sufficient to correctly spread the differences in the LBlock key-schedule.

However, the four related key differential paths given in Appendix A do not work for all possible values of the bits $K_{75}, K_{74}, K_{73}, K_{72}$ of the key. But, from those four related key paths, we are able to have a complete partition of all possible values. This is due to the small size of the S-boxes that work on nibbles. Moreover, those differentials cross almost always the same S-box s_8 leading to always the same differences.

Thus, it will be always possible according to the value of 5 bits (see details in Appendix A) of the master key K to build a second key $K' = K \oplus \Delta K$ with ΔK equal to 0 everywhere except on the nibble $K_{75}, K_{74}, K_{73}, K_{72}$ which takes the value 2 or 4.

3.2. Impossible differential path

In the original paper describing LBlock, the authors give the following 14-round impossible differential:

$$(00000000, 00\alpha 00000)$$

$$\text{after 14 rounds could not give } (0\beta 000000, 00000000).$$

As the differences injected through the subkey additions in our related key sets have really low weight, we are able to continue to construct 14-round and 15-round impossible differentials even taking into account the differences coming from the subkeys. For example, the following 15-round impossible differential (starting at the beginning of round 5 and ending after round 19) cannot happen:

$$(00000000, 0000000\alpha)$$

$$\text{after 15 rounds cannot give } (00000000, 00000000).$$

The fact that the output difference could be completely null comes from the injection of differences coming from the subkey additions.

The complete details of this impossible differential are given in Fig. 2. This impossible differential works for all the related key paths presented in Appendix A. As we just said, this impossible differential is taken from the fifth round until the 19th round and combined with the first four rounds at the beginning as shown on Fig. 3 and with the last four rounds in the end as shown on Fig. 4.

3.3. The attack description for 22 rounds

If we consider Figs. 3 and 4 we see which differences will have the extended impossible differential path in the first and in the last round. In Fig. 3 we show a case that

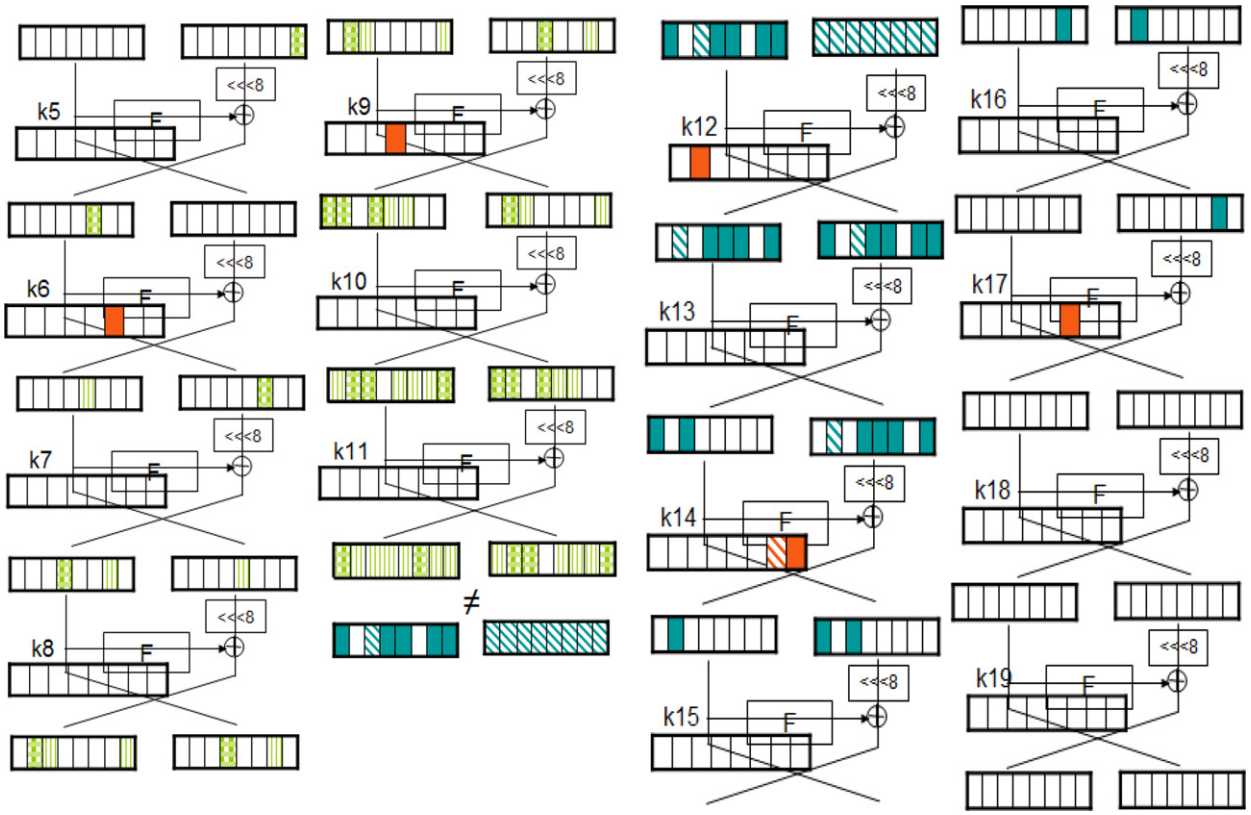

Fig. 2. The impossible differential used in the attack. Nibbles marked in grey symbolize some non-zero differences in the subkeys and in the backward sens. Checkerboard hashed nibbles symbolize some non-zero differences in direct sens. Vertically hashed nibbles mean that differences could be null in direct sens. Diagonally hashed nibbles are for differences that could be null in the backward direction.

works for 2 out of the 4 possible differential paths, being similar for the other two.

The procedure of our attack is as follows:

- For each one of the 4 possible differential paths in the key schedule, we find m good pairs of input messages that satisfy the extended differential path. This can be done by the limited-birthday approach with a complexity of about $m \cdot 2^{12} 2^{32-23} = m \cdot 2^{21}$ as the size of differences in the output is of 12 bits. As the partial keybits will be determined only in a second step, we need to build the m set and repeat the following procedure for all the 4 possibilities of the differential path in the key schedule.
- For each of the m good pairs (and for the 4 possible differential paths) we check if the conditions of getting from the input pair to the beginning of the impossible differential, and from the output to the end of the impossible differential, can be verified by some values of the keybits that intervene in these conditions. In total, we have 57 keybits involved.
- The keybits that make both transitions possible for at least one of the m good pairs will be filtered out of the possible key guesses as otherwise they would imply that the impossible differential had occurred. We will compute next which size must m have so that we filter all the wrong key guesses.
- From Fig. 3, we can see that there are 7 nibble conditions for erasing the active nibbles and obtaining the differential configuration at the input of the impossible differential. The involved keybits are K_{77} to K_{68}, K_{63} to K_{48}, K_{46} to K_{41}, K_{34} to K_{31}, K_{26} to K_{19} (44 in total).
- From Fig. 4, we can see that there exist 3 nibble conditions for obtaining from the output, the differential configuration of the end of the impossible differential. They involve keybits K_{76} to K_{73}, K_{55} to K_{52}, K_{47} to K_{44}, K_{30} to K_{27}, K_{18} to K_{15} and K_6 to K_3 (24 in total and just 13 not included in the previous set).
- As the probability that for a good pair, the $7+3=10$ nibble conditions are verified is 2^{-40}, for each key guess the probability that none of the m good pairs verifies all the conditions is

$$P = \left(1 - 2^{-40}\right)^m.$$

- We have 2^{57} possibilities for the involved keybits, which means that if we choose $m = 2^{47}$, and so $P \approx 2^{-184.66}$, we will filter out all the wrong key guesses but the correct one. We can expect that $2^{-184.66+57} = 2^{-127.66}$ wrong guesses remain. So, we find the correct key with a very high probability.

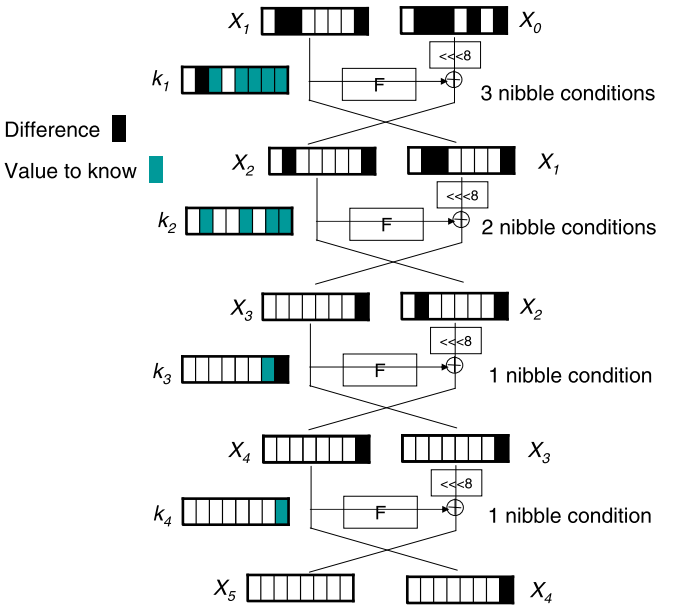

Fig. 3. The initial rounds.

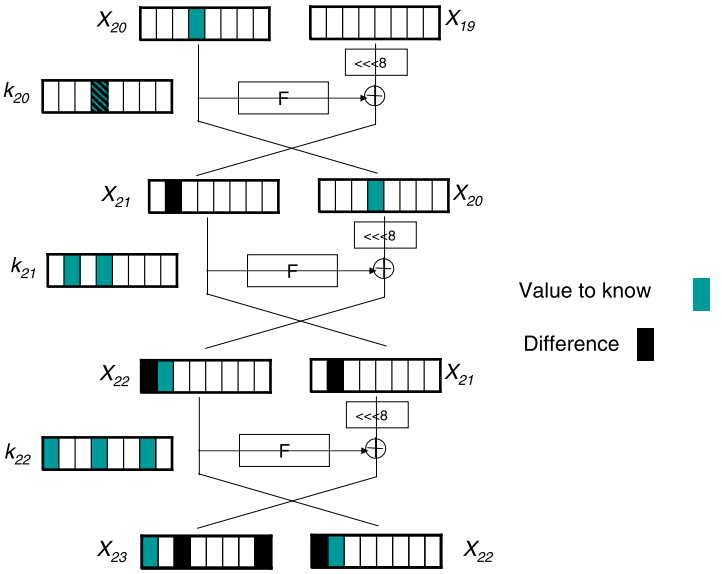

Fig. 4. The final rounds.

The complexity of the attack, where we recover 57 keybits (and then the remaining ones with much lower complexity) is then

$$4 \cdot 2^{47+21} + 4 \cdot 2^{47} 2^{57-40} \approx 2^{70},$$

where the first term represents the complexity of obtaining the 2^{47} pairs with the wanted input–output differences for the 4 differential paths of the key schedule, and the second term comes from the fact that, for each of the 2^{47} pairs of messages, and for the 4 possible key schedule paths, we filter out all the partial keys that verify the conditions. For each one of the m pairs, we have on average $2^{57-40} = 2^{13}$ such values that verify the 10 nibble conditions for the 57 keybits.

4. Conclusion

We have provided in this letter a more detailed analysis of related key impossible differential behaviors of the new lightweight block cipher LBlock. We take advantage in the proposed related key impossible differential attack of some particular weaknesses of the key-schedule that could produce differential paths with really low weight for an initial difference carefully chosen. The complexity

ΔK:	020000000000 00000000	ΔK:	040000000000 00000000	ΔK:	020000000000 00000000	ΔK:	040000000000 00000000
ΔSK_1:	0 2 0 0 0 0 0 0	ΔSK_1:	0 4 0 0 0 0 0 0	ΔSK_1:	0 2 0 0 0 0 0 0	ΔSK_1:	0 4 0 0 0 0 0 0
ΔSK_2:	0 0 0 0 0 0 0 0	ΔSK_2:	0 0 0 0 0 0 0 0	ΔSK_2:	0 0 0 0 0 0 0 0	ΔSK_2:	0 0 0 0 0 0 0 0
ΔSK_3:	0 0 0 0 0 0 0 8	ΔSK_3:	0 0 0 0 0 0 1 0	ΔSK_3:	0 0 0 0 0 0 0 8	ΔSK_3:	0 0 0 0 0 0 1 0
ΔSK_4:	0 0 0 0 0 0 0 0	ΔSK_4:	0 0 0 0 0 0 0 0	ΔSK_4:	0 0 0 0 0 0 0 0	ΔSK_4:	0 0 0 0 0 0 0 0
ΔSK_5:	0 0 0 0 0 0 0 0	ΔSK_5:	0 0 0 0 0 0 0 0	ΔSK_5:	0 0 0 0 0 0 0 0	ΔSK_5:	0 0 0 0 0 0 0 0
ΔSK_6:	0 0 0 0 0 4 0 0	ΔSK_6:	0 0 0 0 0 8 0 0	ΔSK_6:	0 0 0 0 0 4 0 0	ΔSK_6:	0 0 0 0 0 8 0 0
ΔSK_7:	0 0 0 0 0 0 0 0	ΔSK_7:	0 0 0 0 0 0 0 0	ΔSK_7:	0 0 0 0 0 0 0 0	ΔSK_7:	0 0 0 0 0 0 0 0
ΔSK_8:	0 0 0 0 0 0 0 0	ΔSK_8:	0 0 0 0 0 0 0 0	ΔSK_8:	0 0 0 0 0 0 0 0	ΔSK_8:	0 0 0 0 0 0 0 0
ΔSK_9:	0 0 0 2 0 0 0 0	ΔSK_9:	0 0 0 4 0 0 0 0	ΔSK_9:	0 0 0 2 0 0 0 0	ΔSK_9:	0 0 0 4 0 0 0 0
ΔSK_{10}:	0 0 0 0 0 0 0 0	ΔSK_{10}:	0 0 0 0 0 0 0 0	ΔSK_{10}:	0 0 0 0 0 0 0 0	ΔSK_{10}:	0 0 0 0 0 0 0 0
ΔSK_{11}:	0 0 0 0 0 0 0 0	ΔSK_{11}:	0 0 0 0 0 0 0 0	ΔSK_{11}:	0 0 0 0 0 0 0 0	ΔSK_{11}:	0 0 0 0 0 0 0 0
ΔSK_{12}:	0 6 0 0 0 0 0 0	ΔSK_{12}:	0 2 0 0 0 0 0 0	ΔSK_{12}:	0 2 0 0 0 0 0 0	ΔSK_{12}:	0 6 0 0 0 0 0 0
ΔSK_{13}:	0 0 0 0 0 0 0 0	ΔSK_{13}:	0 0 0 0 0 0 0 0	ΔSK_{13}:	0 0 0 0 0 0 0 0	ΔSK_{13}:	0 0 0 0 0 0 0 0
ΔSK_{14}:	0 0 0 0 0 0 1 8	ΔSK_{14}:	0 0 0 0 0 0 0 8	ΔSK_{14}:	0 0 0 0 0 0 0 8	ΔSK_{14}:	0 0 0 0 0 0 1 8
ΔSK_{15}:	0 0 0 0 0 0 0 0	ΔSK_{15}:	0 0 0 0 0 0 0 0	ΔSK_{15}:	0 0 0 0 0 0 0 0	ΔSK_{15}:	0 0 0 0 0 0 0 0
ΔSK_{16}:	0 0 0 0 0 0 0 0	ΔSK_{16}:	0 0 0 0 0 0 0 0	ΔSK_{16}:	0 0 0 0 0 0 0 0	ΔSK_{16}:	0 0 0 0 0 0 0 0
ΔSK_{17}:	0 0 0 0 0 c 0 0	ΔSK_{17}:	0 0 0 0 0 4 0 0	ΔSK_{17}:	0 0 0 0 0 4 0 0	ΔSK_{17}:	0 0 0 0 0 c 0 0
ΔSK_{18}:	0 0 0 0 0 0 0 0	ΔSK_{18}:	0 0 0 0 0 0 0 0	ΔSK_{18}:	0 0 0 0 0 0 0 0	ΔSK_{18}:	0 0 0 0 0 0 0 0
ΔSK_{19}:	0 0 0 0 0 0 0 0	ΔSK_{19}:	0 0 0 0 0 0 0 0	ΔSK_{19}:	0 0 0 0 0 0 0 0	ΔSK_{19}:	0 0 0 0 0 0 0 0
ΔSK_{20}:	0 0 0 6 0 0 0 0	ΔSK_{20}:	0 0 0 2 0 0 0 0	ΔSK_{20}:	0 0 0 2 0 0 0 0	ΔSK_{20}:	0 0 0 6 0 0 0 0
ΔSK_{21}:	0 0 0 0 0 0 0 0	ΔSK_{21}:	0 0 0 0 0 0 0 0	ΔSK_{21}:	0 0 0 0 0 0 0 0	ΔSK_{21}:	0 0 0 0 0 0 0 0
ΔSK_{22}:	0 0 0 0 0 0 0 0	ΔSK_{22}:	0 0 0 0 0 0 0 0	ΔSK_{22}:	0 0 0 0 0 0 0 0	ΔSK_{22}:	0 0 0 0 0 0 0 0
ΔSK_{23}:	0 5 0 0 0 0 0 0	ΔSK_{23}:	0 3 0 0 0 0 0 0	ΔSK_{23}:	0 2 0 0 0 0 0 0	ΔSK_{23}:	0 1 0 0 0 0 0 0
ΔSK_{24}:	0 0 0 0 0 0 0 0	ΔSK_{24}:	0 0 0 0 0 0 0 0	ΔSK_{24}:	0 0 0 0 0 0 0 0	ΔSK_{24}:	0 0 0 0 0 0 0 0
ΔSK_{25}:	0 0 0 0 0 0 1 4	ΔSK_{25}:	0 0 0 0 0 0 0 c	ΔSK_{25}:	0 0 0 0 0 0 0 8	ΔSK_{25}:	0 0 0 0 0 0 0 4
ΔSK_{26}:	0 0 0 0 0 0 0 0	ΔSK_{26}:	c 0 0 0 0 0 0 0	ΔSK_{26}:	0 0 0 0 0 0 0 0	ΔSK_{26}:	c 0 0 0 0 0 0 0
ΔSK_{27}:	0 0 0 0 0 0 0 0	ΔSK_{27}:	0 0 0 0 0 0 0 0	ΔSK_{27}:	0 0 0 0 0 0 0 0	ΔSK_{27}:	0 0 0 0 0 0 0 0
ΔSK_{28}:	0 0 0 0 0 b 4 0	ΔSK_{28}:	0 0 0 0 0 7 0 0	ΔSK_{28}:	0 0 0 0 0 4 0 0	ΔSK_{28}:	0 0 0 0 0 2 c 0
ΔSK_{29}:	0 0 0 0 0 0 0 0	ΔSK_{29}:	0 0 0 0 0 0 0 0	ΔSK_{29}:	0 0 0 0 0 0 0 0	ΔSK_{29}:	0 0 0 0 0 0 0 0
ΔSK_{30}:	0 0 0 0 0 0 0 0	ΔSK_{30}:	0 0 0 0 0 0 0 0	ΔSK_{30}:	0 0 0 0 0 0 0 0	ΔSK_{30}:	0 0 0 0 0 0 0 0
ΔSK_{31}:	0 0 0 5 a 0 0 0	ΔSK_{31}:	0 0 0 3 8 0 0 0	ΔSK_{31}:	0 0 0 2 0 0 0 0	ΔSK_{31}:	0 0 0 1 6 0 0 0
ΔSK_{32}:	0 0 0 0 0 0 0 0	ΔSK_{32}:	0 0 0 0 0 0 0 0	ΔSK_{32}:	0 0 0 0 0 0 0 0	ΔSK_{32}:	0 0 0 0 0 0 0 0

Fig. 5. The four related key differentials used in the attack presented in Section 3 that provide a complete partition on all possible key values. Note that the differential trails from Subkey23 and Subkey26 respectively could have different values from the ones given here.

of the described attack is 2^{70} cipher operations requiring 2^{47} plaintexts.

Finally, we have been able to give the best attack known on LBlock, that works up to 22 rounds, while the analysis for the biggest number of rounds in the original article worked on 20 rounds. We believe that our analysis can still be improved because the overall complexity is far from the cost of the complete exhaustive key search.

Appendix A. Related key differences used in the related key impossible differential attack

There are four cases of related key differentials that depend on the value of the five bits K_{76} and $(K_{75}, K_{74}, K_{73}, K_{72})$. According to the values of those bits, the difference that must be injected in the key is 2 or 4 on K_{18}. The table in Fig. 5 give those differentials on the keys and on the subkeys. We then obtain 4 related key differentials that could be used whatever the 5 bits values are. The four differences are chosen according to:

- If the key bits $(K_{75}, K_{74}, K_{73}, K_{72})$ take the values 0, 1, 4 and 5 and if $K_{76} = 0$, then the good related key differential is the second one given in Fig. 5, else if $K_{76} = 1$, the good related key differential is the fourth one given in Fig. 5.
- If the key bits $(K_{75}, K_{74}, K_{73}, K_{72})$ take the values 2, 3, 6 and 7 and if $K_{76} = 0$, then the related key differential is the fourth one given in Fig. 5, else if $K_{76} = 1$, the good related key differential is the second one given in Fig. 5.
- If the key bits $(K_{75}, K_{74}, K_{73}, K_{72})$ take the values 8, 9, 10 and 11 and if $K_{76} = 0$, then the related key differential is the first one given in Fig. 5, else if $K_{76} = 1$, the good related key differential is the third one given in Fig. 5.
- If the key bits $(K_{75}, K_{74}, K_{73}, K_{72})$ take the values 12, 13, 14 and 15 and if $K_{76} = 0$, then the related key differential is the third one given in Fig. 5, else if $K_{76} = 1$, the good related key differential is the first one given in Fig. 5.

As shown in Fig. 5, the key schedule algorithm does not provide a sufficient diffusion of differences.

References

[1] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J.B. Robshaw, Yannick Seurin, C. Vikkelsoe,

PRESENT: An ultra-lightweight block cipher, in: Cryptographic Hardware and Embedded Systems – CHES 2007, in: LNCS, vol. 4727, Springer, 2007, pp. 450–466.
[2] Andrey Bogdanov, Christian Rechberger, A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN, in: Selected Areas in Cryptography – SAC 2010, in: LNCS, vol. 6544, Springer, 2010, pp. 229–240.
[3] Christophe De Cannière, Orr Dunkelman, Miroslav Knezevic, KATAN and KTANTAN – A family of small and efficient hardware-oriented block ciphers, in: Cryptographic Hardware and Embedded Systems – CHES 2009, in: LNCS, vol. 5747, Springer, 2009, pp. 272–288.
[4] Zhen Gong, Svetla Nikova, Yee-Wei Law, KLEIN: a new family of lightweight block ciphers, in: RFIDSec, 2011.
[5] Jian Guo, Thomas Peyrin, Axel Poschmann, Matthew J.B. Robshaw, The LED block cipher, in: Cryptographic Hardware and Embedded Systems – CHES 2011, in: LNCS, vol. 6917, Springer, 2011, pp. 326–341, http://dx.doi.org/10.1007/978-3-642-23951-9_22.
[6] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, Seongtaek Chee, HIGHT: A new block cipher suitable for low-resource device, in: Cryptographic Hardware and Embedded Systems – CHES 2006, in: LNCS, vol. 4249, Springer, 2006, pp. 46–59.
[7] Gregor Leander, Christof Paar, Axel Poschmann, Kai Schramm, New lightweight DES variants, in: Fast Software Encryption – FSE 2007, in: LNCS, vol. 4593, Springer, 2007, pp. 196–210.
[8] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, Erik Zenner, A cryptanalysis of printcipher: The invariant subspace attack, in: Advances in Cryptology – CRYPTO 2011, in: LNCS, vol. 6841, Springer, 2011, pp. 206–221.
[9] Marine Minier, María Naya-Plasencia, Some preliminary studies on the differential behavior of the lightweight block cipher lblock, in: ECRYPT Workshop on Lightweight Cryptography, November 28–29, Louvain-la-Neuve, Belgium, 2011.
[10] Matthew J.B. Robshaw, Searching for compact algorithms: CGEN, in: VIETCRYPT, 2006, pp. 37–49.
[11] François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, Jean-Jacques Quisquater, SEA: A scalable encryption algorithm for small embedded applications, in: CARDIS, 2006, pp. 222–236.
[12] Wenling Wu, Lei Zhang, Lblock: A lightweight block cipher, in: Applied Cryptography and Network Security – ACNS 2011, in: LNCS, vol. 6715, Springer, 2011, pp. 327–344.