Truncated Differential Analysis of Reduced-Round LBlock*

Sareh Emami[1,2], Cameron McDonald[2], Josef Pieprzyk[1], and Ron Steinfeld[3]

[1] Macquarie University, Australia
[2] Qualcomm Incorporated, Australia
[3] Monash University, Australia

Abstract. In this paper we present truncated differential analysis of reduced-round LBlock by computing the differential distribution of every nibble of the state. LLR statistical test is used as a tool to apply the distinguishing and key-recovery attacks. To build the distinguisher, all possible differences are traced through the cipher and the truncated differential probability distribution is determined for every output nibble. We concatenate additional rounds to the beginning and end of the truncated differential distribution to apply the key-recovery attack. By exploiting properties of the key schedule, we obtain a large overlap of key bits used in the beginning and final rounds. This allows us to significantly increase the differential probabilities and hence reduce the attack complexity. We validate the analysis by implementing the attack on LBlock reduced to 12 rounds. Finally, we apply single-key and related-key attacks on 18 and 21-round LBlock, respectively.

Keywords: Block cipher, LBlock, Truncated differetial analysis, Probability distribution, Log-likelihood ratio, Key-recovery attack.

1 Introduction

With the advent of RFID technology in communication applications, traditional block ciphers are generally not suitable for resource constrained devices. Lightweight block ciphers (with smaller block and key size) are a new class of ciphers designed for such environments. Recently there have been a lot of new lightweight designs, examples include: HIGHT [8], PRESENT [5], PRINTcipher [9], and LBlock [17]. Security analysis of lightweight primitives is currently receiving considerable attention.

Similarly to the other lightweight block ciphers, LBlock has attracted a significant amount of cryptanalysis. For instance, related-key impossible differential attacks were successfully applied to 21 and 22-round LBlock [13,14]. A 16-round related-key truncated differential is exploited to launch an attack on 22-round LBlock [12]. In [15], a 15-round distinguisher is proposed, allowing an integral

* Sareh Emami is supported by the Macquarie University MQRES scholarship. Josef Pieprzyk and Ron Steinfeld are supported by the ARC grant DP0987734. Ron Steinfeld is also supported by ARC Australian Research Fellowship (ARF).

M. Abdalla, C. Nita-Rotaru, and R. Dahab (Eds.): CANS 2013, LNCS 8257, pp. 291–308, 2013.

attack for up to 22 rounds. Zero-correlation linear cryptanalysis of 22-round LBlock is presented in [16]. All attacks published so far require high amount of memory and data.

The standard differential analysis and its derivatives usually follow a differential trail and compute probabilities for known expected differences. Recently differential distribution analysis got high attention in the analysis of block ciphers. These type of attacks typically require lower amount of data in comparision to the standard differential. In the case of lightweight block ciphers, Albrecht and Leander explained in [1], that it is feasible to find the probability distribution of all output differences from one (or more) input difference. In a similar work, multiple differential cryptanalysis using the LLR and χ^2 statistical tests discussed in [3]. However in [1,3] the differential distribution is found for the whole state, which makes the attack possible only on a cipher with a *small* block size. The link between differential analysis and correlations of linear approximations, was exploited in [4] to compute truncated differential probabilities. This method combined with LLR test used to apply multiple differential cryptanalysis on PRESENT.

In this paper we present the truncated differential analysis of LBlock by looking at the difference distributions of the state nibbles independently. After finding a distribution that significantly differs from that of a random permutation, we use LLR statistical test to build the distinguisher. The way we find the truncated differential distribution in the markov model, makes our attack possible on the ciphers with relatively larger states than [1,3]. Additional rounds are added to the end of the distinguisher to be used in a partial key recovery phase. Moreover, by exploiting related key bits in the key schedule, we concatenate additional rounds to the beginning of the distinguisher. Differentials through these beginning rounds have high probability, allowing us to extend the attack without significantly increasing the complexity. We apply the attack on a reduced round LBlock and construct single key and related key attacks up to 18 and 21 rounds, respectively. A comparison with attack complexities from prior work is given in Table 1.

The rest of the paper is structured as follows. Preliminaries are explained in Section 2. A framework to apply the key-recovery attack using the truncated differential distribution, while benefiting the key schedule properties is introduced in Section 3. Section 4 discusses the complexity of the attack and includes the empirical results. Section 5 presents a single-key attack on 18 rounds as well as related-key attacks on 20 and 21 rounds of LBlock. Finally, we conclude the paper in Section 6.

2 Preliminaries

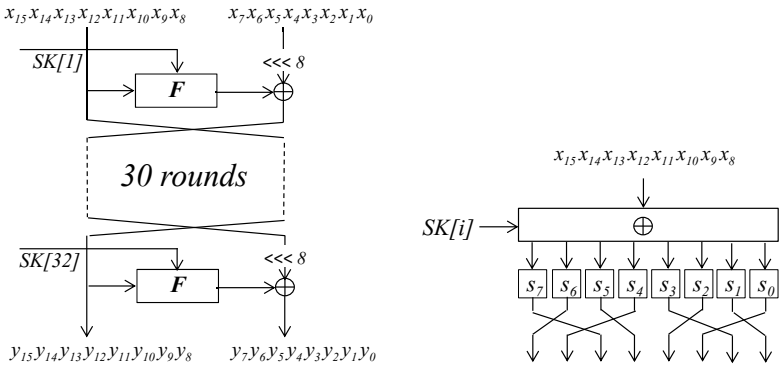
2.1 LBlock Description

LBlock [17] is a lightweight block cipher with a block size of 64 bits and a key size of 80 bits. The design is a 32 round balanced Feistel where the input block is divided into two 32-bit halves, denoted the *left-hand* half (most significant

Table 1. Attacks on LBlock

Type of Attack	rounds	Data	Time	Reference
Related-key impossible differential	22	2^{68}	2^{70}	[14]
Related-key differential	22	$2^{64.1}$	2^{67}	[12]
Integral	18	$2^{62} + 2^{20}$ memory	2^{36}	[17]
Integral	22	$2^{61} + 2^{63}$ memory	2^{70}	[15]
Zero-correlation linear	22	$2^{60} + 2^{64}$ memory	2^{79}	[16]
Truncated differential	18	2^{23}	$2^{68.71}$	This paper
Related-key truncated differential	20	2^{27}	$2^{74.55}$	This paper
Related-key truncated differential	21	2^{30}	$2^{77.56}$	This paper

bits) and the *right-hand* half (least significant bits). Each round includes a key addition, where the round sub-keys are 32-bit values denoted by $SK[i]$. The structure of LBlock is shown in Fig. 1a.



(a) Feistel network (b) The round function

Fig. 1. LBlock structure

The round function includes a XOR key addition, a nonlinear S-box layer (S) and a linear permutation layer (P). The S-box layer S applies 8 different S-boxes (s_i) in parallel. The linear layer P simply reorders the 8 nibbles in the state. The round function is show in Fig. 1b. Since all the state functions operate on 4 bits, it is convenient to represent the state as a sequence of nibbles using the following notation $\boldsymbol{x} = (x_{15}, \ldots, x_1, x_0)$. LBlock uses a key scheduling function to expand the 80 bit master key K into 32 round sub-keys $SK[i]$, each being 32 bits in size. The master key K is stored in a register, denoted by the sequence of

bits $k_{79}k_{78}k_{77}k_{76}\ldots k_1k_0$. The key register is updated by the scheduling process and the 32 most significant bits of the register become the round sub-key. The key scheduling process is as follows:
For $i = 1, 2, \ldots, 31$:

1. $K \lll 29$
2. $[k_{79}k_{78}k_{77}k_{76}] = s_9[k_{79}k_{78}k_{77}k_{76}]$ and $[k_{75}k_{74}k_{73}k_{72}] = s_8[k_{75}k_{74}k_{73}k_{72}]$
3. $[k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus [i]_2$
4. Output the leftmost 32 bits of the current content of register K as the round sub-key $SK[i+1]$.

where s_8 and s_9 are two 4-bit S-boxes.

2.2 Likelihood Test

Let $P = (p_0, p_1, \ldots, p_n)$ and $Q = (q_0, q_1, \ldots, q_n)$ denote two discrete probability distributions of random variables X and Y, respectively. The relative entropy, or Kullback-Leibler divergence, is a measure between two distributions, see [2,6].

Definition 1. *The Kullback-Leibler (KL) divergence between P and Q is defined as follows:*

$$D(P||Q) = \sum_{i=0}^{n} p_i \cdot ln(\frac{p_i}{q_i}) \tag{1}$$

As in [6], we use the convention that $0 \cdot log\frac{0}{q} = 0$ and $p \cdot log\frac{p}{0} = \infty$.

In the *binary hypothesis testing problem*, one is given a set of empirical data $x = (x_0, x_1, \ldots, x_n)$ taken from N samples. The empirical probability distribution is equal to $\hat{P} = (\hat{p}_0, \hat{p}_1, \ldots, \hat{p}_n) = 1/N \cdot (x_0, x_1, \ldots, x_n)$. According to the *Neyman-Pearson Lemma*, the log-likelihood ratio is the optimal method for determining if the sample data belongs to one of two different probability distributions P or Q, see [6,7].

Definition 2. *The log-likelihood ratio (LLR) is defined as*

$$LLR(\hat{P}, P, Q) = N \sum_{i=0}^{n} \hat{p}_i \cdot ln(\frac{p_i}{q_i}) \tag{2}$$

If $LLR(\hat{P}, P, Q) \geq \Theta$ (Θ is a threshold parameter), the empirical data is accepted as a sample from the distribution P (rejecting Q as the hypothesis). Otherwise, P is rejected in favour of Q. In our analysis, we use this to distinguish between distributions representing the *right* key and the *wrong* keys which is explained in later sections.

3 Truncated Differential Analysis

The analysis is structured in to the following three phases: Standard Differential phase (SD), Truncated Differential Distribution (TDD), and Partial-Key Recovery phase (PKR). Fig. 2 depicts the range of each phase. SD phase starts from state S_0 with a known input difference $\boldsymbol{\alpha}$, and follows a standard differential trail through SD-rounds up to state S_1 with specific output difference $\boldsymbol{\beta}$. TDD phase calculates the truncated differential distribution from input $\boldsymbol{\beta}$ through TDD-rounds to state S_2 with output $\boldsymbol{\Gamma}$. The output $\boldsymbol{\Gamma}$ here is not a specific difference but a probability distribution over all possible differences. PKR phase involves partial decryption of the ciphertext to determine S_2 from the observed output state S_3. The difference in state S_2 is measured and compared against the expected distribution $\boldsymbol{\Gamma}$.

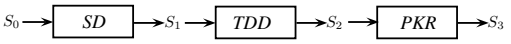


Fig. 2. The attack model

3.1 Standard Differential Phase

The Standard Differential (SD) phase involves finding a high probability differential characteristic through some number of rounds. The XOR-difference between two states $\boldsymbol{x}$ and $\boldsymbol{x}'$ is denoted by $\boldsymbol{\alpha} = (\alpha_{15} \dots \alpha_1 \alpha_0) = (x_{15} \oplus x'_{15}, \dots, x_1 \oplus x'_1, x_0 \oplus x'_0)$. Note that α_i represents exact difference of 4-bits, hence $\alpha_i \in \{0, \dots, 15\}$. The differential trail maps a specific input difference $\boldsymbol{\alpha}$ to a specific output difference $\boldsymbol{\beta}$ with probability denoted $\mathbf{P_{SD}}(\boldsymbol{\alpha} \rightarrow \boldsymbol{\beta})$.

For example, let the input difference be $\boldsymbol{\alpha} = (10000000\ \ 00002000)$. A possible output difference, after one round, is $\boldsymbol{\beta} = (00000000\ \ 10000000)$. The probability of this differential is 2^{-2}.

$$\mathbf{SD} : (10000000\ \ 00002000) \rightarrow (00000000\ \ 10000000) \tag{3}$$

The probability is computed under the assumption that the input values of S-box s_7 are not known. If the inputs to the S-box are known, we can detect (with probability 1) whether the differential trail is followed. This requires knowledge of nibble 7 of $SK[0]$. Conversely, given the values of the state, we can find solutions to the sub-key $SK[0]_7$ such that the differential trail is followed.

3.2 Truncated Differetial Distribution Phase

In this phase, we model the difference distribution of *all* possible output differences for every nibble based on a chosen distribution of input differences. This generalisation is the fundamental idea behind truncated differential analysis [10] and all-in-one differential analysis [1].

Computing Truncated Differential Distribution. The round function consists of two components that affect the probability distribution, S-box transformation and XOR addition. Proposition 1, describes probability of differences for each nibble after an S-box transformation, and Proposition 2 shows how XOR addition affect the difference probability distribution.

Proposition 1. *For an S-box* $s_n : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$ *and input difference probability distribution* $\boldsymbol{x} = (x^0 x^1 \dots x^{15})$*, where* x^i *is the probability of difference* i *for nibble* n*, the output difference probability* y^i *after S-box transformation* s_n *is calculated as*

$$y^i = \sum_{j=0}^{15} x^j \cdot \mathbf{P}(s_n(j) = i) \tag{4}$$

Proof. Assume the difference J occurs with probability x^J and 4-bit S-box s_n transfers difference J to difference I with probability $\mathbf{P}(s_n(J) = I)$. Hence, difference I happens from input diffrence J with probability $x^J \cdot \mathbf{P}(s_n(J) = I)$. However, difference I might occur from s-box transformation of the other 15 input differences; therefore output difference I happens with probability y^I as $y^I = \sum_{j=0}^{15} x^j \cdot \mathbf{P}(s_n(j) = I)$. The same way is used to calculate probability y^i for every output difference $0 \leq i \leq 15$. □

Proposition 2. *For two input difference probabilities* $\boldsymbol{x} = (x^0 x^1 \dots x^{15})$ *and* $\boldsymbol{y} = (y^0 y^1 \dots y^{15})$*, the output XOR-difference probability* z^i *is*

$$z^i = \sum_{j=0}^{15} x^j \cdot y^{i \oplus j} \tag{5}$$

Proof. Assume nibble Z is the XOR-additoion of nibbles X and Y. Difference J at nibble X happens with probability x^J; while, in nibble Y, difference $K = I \oplus J$ happens with probability y^K. By XORing differences J and K, nibble Z has difference I with probability $z^I = x^J \cdot y^{I \oplus J}$. However, difference I might be the result of XORing other 15 differences $0 \leq j \leq 15$ of nibble X with difference $k = I \oplus j$ of nibble Y. Thus, overall difference I happens with probability $z^I = \sum_{j=0}^{15} x^j \cdot y^{I \oplus j}$. For every difference $0 \leq i \leq 15$ of nibble Z probability z^i is calculated with the same way. □

These propositions allow us to construct the differential transformation matrix for the round function; and, given an input distribution, obtain the output truncated differential distribution after a number of rounds. Thus, the TDD phase maps a difference vector $\boldsymbol{\beta}$ to a distribution matrix $\boldsymbol{\Gamma}$. We denote the probability distribution matrix $\mathbf{P_{TDD}}(\boldsymbol{\beta} \rightarrow \boldsymbol{\Gamma})$. For example, let the input difference vector be $\boldsymbol{\beta} = (00000000\ \ 10000000)$. Table 2 lists the output truncated differential distribution $\mathbf{P_{TDD}}(\boldsymbol{\beta} \rightarrow \boldsymbol{\Gamma})$ for the right-hand half nibbles after 8 rounds of LBlock, calculated using Propositions 1 and 2.

The analysis is more effective if a differential distribution profile is chosen in a way that is easiest to distinguish. More specifically, a distribution that

Table 2. Example truncated differential distribution after 8 rounds

Diff\Nibble	7	6	5	4	3	2	1	0
0	0.0610	0.0654	0.0000	0.0000	0.0667	0.0667	0.0000	0.0000
1	0.0000	0.0592	0.0312	0.0693	0.0625	0.0625	0.0625	0.0645
2	0.0649	0.0620	0.1562	0.0732	0.0626	0.0624	0.0312	0.0635
3	0.0649	0.0619	0.0312	0.0684	0.0623	0.0626	0.0938	0.0649
4	0.0610	0.0608	0.0469	0.0698	0.0620	0.0625	0.0625	0.0654
5	0.0732	0.0646	0.0469	0.0610	0.0626	0.0625	0.0625	0.0664
6	0.0703	0.0657	0.0781	0.0649	0.0622	0.0624	0.1250	0.0654
7	0.0684	0.0604	0.1094	0.0698	0.0625	0.0625	0.0625	0.0688
8	0.0703	0.0588	0.0625	0.0635	0.0617	0.0646	0.0625	0.0649
9	0.0679	0.0663	0.0625	0.0649	0.0618	0.0583	0.0625	0.0757
A	0.0659	0.0627	0.0469	0.0635	0.0623	0.0604	0.0312	0.0659
B	0.0649	0.0626	0.0469	0.0728	0.0619	0.0626	0.0312	0.0684
C	0.0615	0.0615	0.0781	0.0659	0.0621	0.0646	0.0625	0.0649
D	0.0679	0.0634	0.1094	0.0654	0.0619	0.0583	0.0625	0.0728
E	0.0693	0.0591	0.0625	0.0620	0.0626	0.0645	0.1250	0.0630
F	0.0684	0.0656	0.0312	0.0654	0.0623	0.0626	0.0625	0.0654
D(P||U)	**6.59e-2**	**7.37e-4**	**1.81e-1**	**6.59e-2**	**1.55e-4**	**5.6e-4**	**1.46e-1**	**6.57e-2**

is significantly different from uniformly random. As described in Section 2.2, KL-divergence is the most accurate way to measure the distance between two distributions [2]. The last row in Table 2 lists the KL-divergence between calculated probability distribution and uniform distribution for every nibble. Here, U denotes the uniform probability distribution with equal probability $\mathbf{P_U} = 1/16$. Note, from Table 2, there are impossible differentials in nibbles 0, 1, 4, 5 and 7. This is due to the short number of rounds used in the sample and does not generally occur in longer trails.

3.3 Partial Key Recovery Phase

Similar to a classical differential attack, additional rounds are added to the end of the truncated differential distinguisher. In this analysis, the method for distinguishing is based on the variance between a differential distribution P and the uniform distribution U. From the truncated differential distribution table, we choose one (or more) nibbles with significantly large KL-divergence. This nibble we term the *target* nibble and set P equal to the probability distribution for this nibble. By guessing a subset of the round keys and decrypting ciphertext pairs through the final rounds, we observe the target nibble differential distribution. For LBlock, it is not required that the entire sub-key be known to determine nibbles from previous rounds. For example, we choose nibble 3

(of the right-hand half) as the target nibble. Table 3 lists the nibbles required to decrypt 3 rounds and determine nibble 3. The X signifies nibbles that must be calculated in order to decrypt back to the target nibble. The master key bits are the key bits used relative to the master encryption key at round $n-3$.

Table 3. Nibbles required to decrypt 3 rounds

Round	Left nibbles	Right nibbles	sub-key nibbles	Master Key bits
n-3	- - - - - - - -	- - - - X - - -	X - - - - - - -	79-78-77-76
n-2	- - X - - - - -	X - - - - - - -	- - - - X - - -	34-33-32-31
n-1	- - - - - - - X	- - X - X - - -	X X - - - - - -	21-20-19-18 17-16-15-14
n	X - X - - - - -	X X - - - - X -	- - - - - - - -	

For every partial key guess, we decrypt N ciphertext pairs and count the frequency of each difference in the target nibble. The difference frequency is stored in an array of 16 counters $\boldsymbol{c} = (c_0 c_1 \dots c_{15})$. The corresponding probability distribution $\hat{P}$ for this sample is $\hat{P} = 1/N \cdot \boldsymbol{c}$ which allows us to calculate the LLR for each key guess. The LLR is used to determine if the observed data most likely belongs to distribution P or U. If P is chosen in favour of U, the guessed key is considered a potential solution for the real key. Otherwise, it is discarded.

3.4 Combining SD and TDD Phases

We can combine the standard differential trail of SD with the truncated differential distribution of TDD to achieve a differential profile over an extended number of rounds. However, the expected output difference probabilities of TDD change due to the success probability of each possible SD differential output. The probability distribution of differences resulting from the input difference α can be computed as follows:

$$\begin{aligned}\mathbf{P_{TDD}}(\boldsymbol{\alpha} \to \boldsymbol{\Gamma}) &= \sum_i \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_i) \cdot \mathbf{P_{TDD}}(\boldsymbol{\beta}_i \to \boldsymbol{\Gamma}_i) \\ &= \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_j) \cdot \mathbf{P_{TDD}}(\boldsymbol{\beta}_j \to \boldsymbol{\Gamma}_j) \\ &+ \sum_{i \neq j} \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_i) \cdot \mathbf{P_{TDD}}(\boldsymbol{\beta}_i \to \boldsymbol{\Gamma}_i),\end{aligned} \tag{6}$$

where $\boldsymbol{\beta_i}$ are all possible output difference vectors of the SD phase. In Equation (6), $\boldsymbol{\beta}_j$ is the input difference for the truncated differential distribution TDD that has the most distinguishable profile (highest KL-divergence). Usually, $\boldsymbol{\beta}_j$ is the difference with the lowest hamming weight. Also, in practice, all other $\boldsymbol{\beta}_i$ lead to probability distributions that are much closer to uniform (in comparison to $\boldsymbol{\beta}_j$). That is,

$$\sum_{i \neq j} \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_i) \cdot \mathbf{P_{TDD}}(\boldsymbol{\beta}_i \to \boldsymbol{\Gamma}_i) \approx (1 - \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_j)) \cdot \mathbf{P_U} \tag{7}$$

From (6) and (7), the output probability distribution is approximated by

$$\mathbf{P_{TDD}}(\boldsymbol{\alpha} \to \boldsymbol{\Gamma}) \approx \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_j) \cdot \mathbf{P_{TDD}}(\boldsymbol{\beta}_j \to \boldsymbol{\Gamma}_j) + (1 - \mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}_j)) \cdot \mathbf{P_U} \tag{8}$$

3.5 Dependencies between SD and PKR Phases

From the key schedule, there is a strong dependency between the sub-key bits guessed in PKR and the sub-key bits affecting SD. This changes the success probability $\mathbf{P_{SD}}$. Note there are two S-boxes s_8 and s_9 used in the key scheduling. These S-boxes introduce nonlinear relationships between sub-keys, meaning the PKR key bits are not always directly obtained from SD key bits. We select the SD and PKR phases in a way such that there are as many common bits as possible for the key bits used in the PKR and SD phases.

3.6 12-Round Example

This section gives details about how the analysis is applied to a 12-round version of LBlock. We construct a 9-round differential distinguisher by combining the 1-round $\mathbf{SD}(\boldsymbol{\alpha} \to \boldsymbol{\beta})$ (from (3)) with 8-round $\mathbf{TDD}(\boldsymbol{\beta} \to \boldsymbol{\Gamma})$ (from Section 3.2). An additional 3 rounds are added for the PKR phase (described in Section 3.3). The entire attack structure is depicted in Fig. 3.

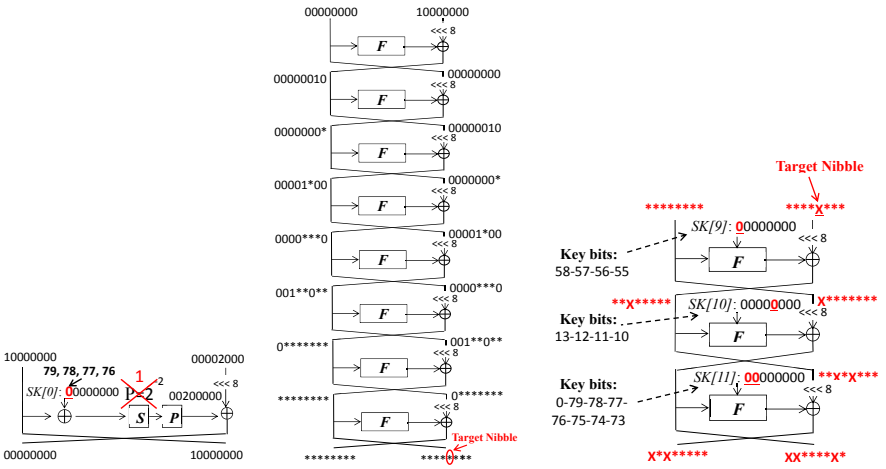
To cover the general application of the analysis, we choose nibble 3 as the target nibble for the PKR phase, which does not benefit from the impossible differential. The sub-keys required to decrypt the ciphertext in the PKR phase (i.e. The underlined sub-key nibbles in Fig. 3c) include $SK[11]_7, SK[11]_6, SK[10]_5$ and $SK[9]_2$, a total of 16 unique bits. The sub-key used in SD phase is $SK[0]_7$. From the key schedule we get

$$SK[0]_7 = ((s_9^{-1}(SK[11]_7) \ \&\ \texttt{0x7}) \ll 1) \mid (s_8^{-1}(SK[11]_6) \ \&\ \texttt{0x1}).$$

That is, for a given guess in PKR phase, we determine the sub-key used in the SD phase.

For a chosen input plaintext pair (with difference $\boldsymbol{\alpha}$), we say it is a *right-pair* if it follows the differential SD. Otherwise, the pair is termed a *wrong-pair*. Note that the attacker does not have access to the internal differential states, he only sees the ciphertext pair. For random input pairs, $\mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}) = 2^{-2}$, and we expect 1/4 right-pairs on average. Henceforth, we denote the total number of plaintext pairs N_p and the number of right-pairs N. For every guess of key bits in PKR, we determine $SK[0]_7$ and distinguish right-pairs from wrong-pairs (with respect to the key guess). By disregarding wrong-pairs we can increase the probability of the SD phase such that $\mathbf{P_{SD}}(\boldsymbol{\alpha} \to \boldsymbol{\beta}) = 1$. Therefore, from Equation (8), $\mathbf{P_{TDD}}(\boldsymbol{\alpha} \to \boldsymbol{\Gamma}) = \mathbf{P_{TDD}}(\boldsymbol{\beta} \to \boldsymbol{\Gamma})$.

When $SK[0]_7$ is incorrect (due to an incorrect guess in PKR), we mistake a wrong-pair for a right-pair. This false-positive results in the addition of noise to the observed probability distribution. The noise is assumed to be uniformly



(a) SD phase (b) TDD phase (c) PKR phase

Fig. 3. 3 phases of the 12-round example

random, a similar assumption to the Wrong Key Randomization Hypothesis [11] (explained later). However, this false-positive only occurs for incorrect guesses and does not affect the correct guess distribution.

4 Complexity Analysis

For each key guessed in the PKR phase, we calculate the LLR between the observed truncated differetial distribution and the expected one. If the LLR is above some threshold (Θ), we consider the guessed key a candidate for the right key. The resulting list of candidate keys are checked for correctness. The attack is successful *if* the right key is among the list of candidate keys, we call this the *attack success rate*. In [1], the "gain" of the attack is the fraction of wrong keys ranked above the expected rank of the right key. We extend this concept and determine the expected number of candidate keys and the effort required to find the right key among them.

Assume R is a random variable for the LLR of the right candidate. After decrypting N pairs of ciphertexts, the expected count for the right candidate is defined by $E(R)$ in Equation (9). Likewise, random variable W is defined for the wrong candidates. The value $E(W)$ gives the expected count of the wrong candidate, defined in Equation (10).

$$E(R) = N \sum_i p_i \, ln(\frac{p_i}{q_i}) \tag{9}$$

$$E(W) = N \sum_i q_i \, ln(\frac{p_i}{q_i}) \tag{10}$$

Here N is the number of right-pairs, p_i is the probability of the expected right key that gives the difference i (which is found in the TDD phase), and q_i is the probability of getting the difference by a wrong key. According to the *Wrong Key Randomization Hypothesis* [11], difference probabilities after decryption by a wrong key candidate are distributed as for a random permutation. Our experiments on LBlock confirm the hypothesis for two or more rounds of decryption.

It is shown in [1] that LLR distribution of the right key is approximated by a normal distribution with a mean of $E(R)$ and variance of $Var(R)$ defined in Equation (11). Likewise, the average distribution of the wrong keys, is approximated by another normal distribution with a mean of $E(W)$ and variance of $Var(W)$, given in Equation (12).

$$Var(R) = N\left(\left(\sum_i p_i\Big(ln(\frac{p_i}{q_i})\Big)^2\right) - \left(\sum_i p_i ln(\frac{p_i}{q_i})\right)^2\right) \tag{11}$$

$$Var(W) = N\left(\left(\sum_i q_i\Big(ln(\frac{p_i}{q_i})\Big)^2\right) - \left(\sum_i q_i ln(\frac{p_i}{q_i})\right)^2\right) \tag{12}$$

To verify the theoretical findings by experiments, we implemented the analysis on the 12-round example of Section 3.6. We ran the analysis 1000 times with $N = 2^{16}$ right-pairs each, and found the LLR distribution for random variables R and W. Note in this example we guess 16 key bits in the PKR phase, therefore there are 2^{16} candidate keys. Fig. 4b shows the LLR distribution for the right key from the experiments. Likewise, Fig. 4a shows the average LLR distribution of all the wrong keys. The theoretical values describing these distributions are, $E(R) = 10.2242$, $E(W) = -10.0356$, $Var(R) = 20.8225$, and $Var(W) = 19.7064$.

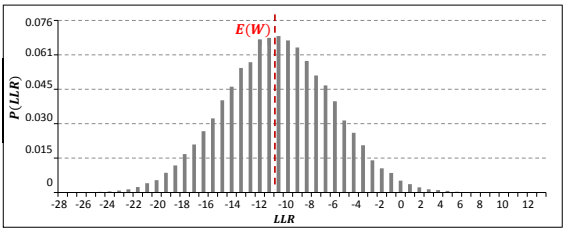
Assume random variable X follows a normal distribution $\mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance, respectively. According to the cumulative distribution function (CDF), the probability of the random variable X falling into the interval $[x, \infty)$ is (erf is the error function of the distribution):

$$\mathbf{P}(X \geq x) = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \tag{13}$$

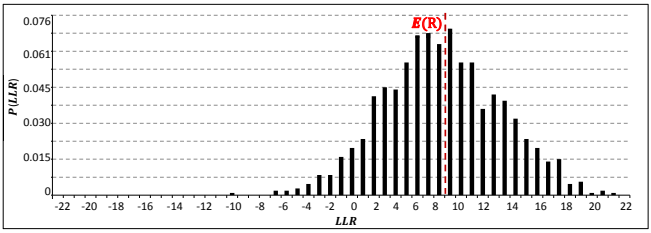
If Θ represents a threshold for the LLR, $\mathbf{P}(R \geq \Theta)$ gives the probability that the right key LLR is greater than the threshold. Likewise, $\mathbf{P}(W \geq \Theta)$ gives the probability of a wrong key LLR greater than the threshold Θ. Both probabilities are calculated from Equation (13). Since $E(R)$ is the mean for the normal distribution of the expected right key, the right key LLR is higher than $E(R)$ with probability $\frac{1}{2}$. While $\mathbf{P}(W \geq E(R))$ gives the probability of a wrong key being ranked higher than the expected right key. If there are N_K key candidates in the test, N_{wk} denotes the wrong keys ranked higher than the threshold. The expected value of N_{wk} is

$$N_{wk} = N_K \cdot \mathbf{P}(W \geq \Theta) \tag{14}$$

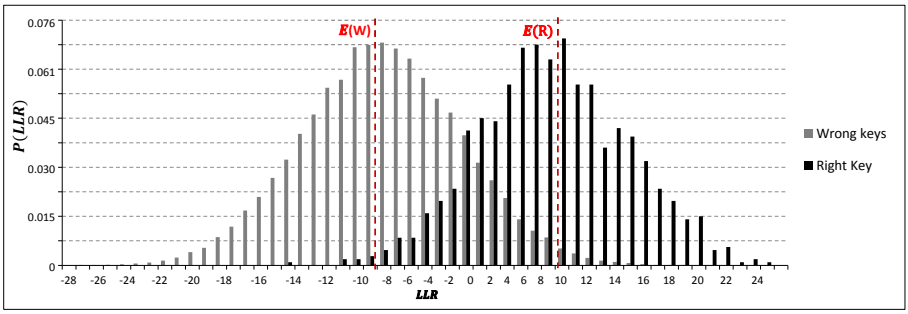
The attack success rate for finding the right key is related to the threshold Θ and N the number of right-pairs (accounting for the SD phase) used in the



(a) Average LLR distribution of the wrong keys



(b) LLR distribution of the right key



(c) Combined diagrams

Fig. 4. Empirical diagrams of the LLR distributions for the 12-round example

attack. By adjusting Θ and N, the attacker is able to find a higher success rate or a lower N_{wk}.

In the 12-round example attack, we choose $N_p = 2^{18}$ chosen plaintext/ciphertext pairs and expect to get $N = 2^{16}$ right-pairs from the SD phase. We ran the experiments 100 times for each chosen threshold. Table 4 shows the results for different success rates by selecting various LLR thresholds. It is clear in Table 4 that the experiments confirm the theory.

After the partial key-recovery, each candidate key should be checked for correctness to do the full key-recovery. One naive method is to guess the remaining unknown key bits by exhaustive search. Assume b_P is the number of PKR-key bits, then the key-recovery attack complexity is

$$C = N\,2^{b_P} + (N_{wk} + 1)\,2^{80-b_P} \tag{15}$$

Table 4. 12-round LBlock results for $N = 2^{16}$ right-pairs

Θ	$\mathbf{P}(R \geq \Theta)$	$\mathbf{P}(W \geq \Theta)$	N_{wk}	Empirical $\mathbf{P}(R \geq \Theta)$	Average empirical N_{wk}
2.6189	0.95	0.0021	143	0.94	154.07
5.6610	0.84	0.0002	14	0.87	15.16
7.1821	0.74	5.25e-05	4	0.73	3.68
8.7032	0.63	1.21e-05	0.79	0.61	0.92
10.2242	0.5	2.51e-06	0.16	0.45	0.19

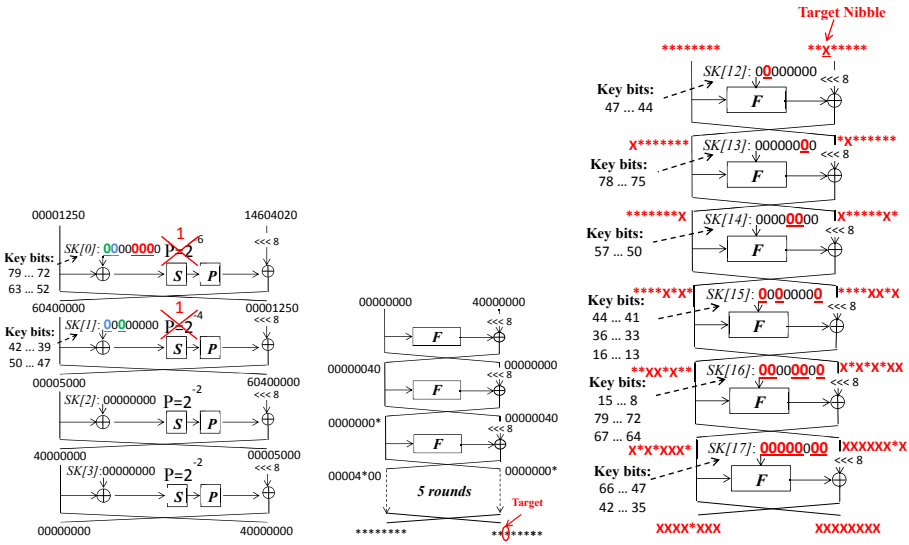
By choosing $N_p = 2^{18}$ plaintext pairs (results in 2^{16} right-pairs) in the 12-round attack, the distinguisher complexity is $2^{16} \times 2^{16} = 2^{32}$. While the whole key recovery attack time complexity is $C = 2^{16} \times 2^{16} + 2^{64} \backsimeq 2^{64}$ encryptions. Note here, exhaustive key search of the remaining bits dominates the complexity. There are more efficient methods for recovering the remaining bits. In cases where the initial phase is the dominant task, the exhaustive search may be used as it does not significantly increase the total complexity.

5 Key-Recovery Attack on LBlock

5.1 Single-Key Attack on 18 Rounds

Fig. 5 describes the truncated differential distribution attack on 18-round LBlock. We divide the 18 rounds into 3 parts to apply the attack. The SD phase takes the first 4 rounds, the TDD phase consists of the next 8 rounds, and the PKR phase includes the 6 final rounds.

The input state of the 4-round SD phase includes 3 nibbles with non-zero differences in the left-hand half and 5 nibbles with non-zero differences in the right-hand half as shown in Fig. 5a. Through the 4-round standard differential almost all the differences are cancelled. So the output state has difference zero in all the nibbles except nibble 7 of the right-hand half. The TDD phase is very similar to that explained in the 12-round attack. It starts with a low weight state (with only difference 4 at nibble 7). Calculating the truncated differential distribution for the right-hand half nibbles at the output state after 8 rounds, the highest KL-divergence occurs with nibble 5 (i.e. $D(P||Q) = 2.184e - 01$). Therefore, nibble 5 is chosen as the target nibble for the 6-round PKR phase. To find the LLR distribution for the target nibble, the attacker needs to guess 52 key-bits in the PKR phase. Observing the SD phase, if the attacker knows the values of 3 sub-key nibbles $SK[0]_1$, $SK[0]_2$ and $SK[0]_3$, he is able to find the output of the 3 active S-boxes in the first round with no extra effort. Likewise, by knowing the values of sub-key nibbles $SK[0]_6$, $SK[0]_7$, $SK[1]_5$ and $SK[1]_7$, he finds the output of 2 active S-boxes in the second round. Overall, he needs to know the values of 28 key bits. These bits are guessed in PKR phase, however going through the key scheduling process the values of bits 73 and 72 are lost.



(a) SD phase (b) TDD phase (c) PKR phase

Fig. 5. Truncated differential distribution attack on 18-round LBlock

By re-guessing these key bits and guessing one more (bit 0), all the required 28 bit values are revealed for the SD phase. Therefore, the probability of the SD phase is increased to $\mathbf{P_{SD}} = 2^{-4}$. As mentioned in Section 3.4, difference probability distribution is updated after combining the SD and TDD phases estimated by Equation (6). Probability distribution of the target nibble 5, is shown in Table 5 before and after combining with the SD phase ($\mathbf{P_T}$ and $\mathbf{P_{ST}}$, respectively).

Adjusting N in Equations (9) and (10), the attacker finds $N = 2^{13}$ as the value with the best trade off between success rate and complexity. The statistical characteristic of the right key and the wrong key distributions are as follows: $E(R) = 6.44$, $E(W) = -6.40$, $Var(R) = 12.99$, and $Var(W) = 12.71$. Table 6 shows the result on 18-round key-recovery attack with different chosen thresholds. Note, the number of plaintext pairs includes those satisfying the first two rounds of the SD phase. Therefore, we need $N_p = 2^{13+10} = 2^{23}$ pairs of plaintext/ciphertext to apply the attack. If the attacker chooses the threshold $\Theta = E(R)$, the probability that he finds the right key is 50% and the attack complexity is $2^{68.71}$.

Table 5. Difference probability distribution of the target nibble

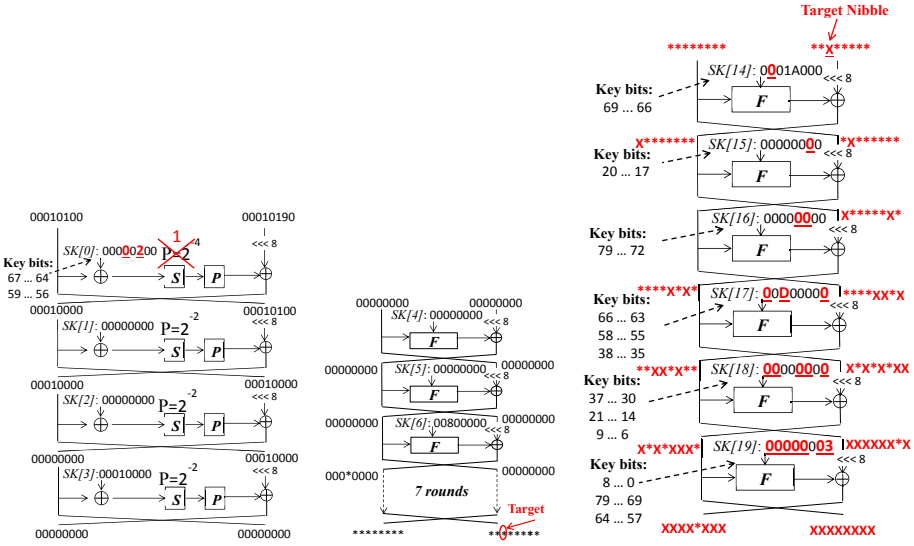
Diff	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\mathbf{P_T}$	0.000	0.156	0.031	0.093	0.046	0.046	0.015	0.109	0.078	0.109	0.031	0.062	0.093	0.031	0.046	0.046
$\mathbf{P_{ST}}$	0.058	0.068	0.060	0.064	0.061	0.061	0.059	0.065	0.063	0.065	0.060	0.062	0.064	0.060	0.061	0.061

Table 6. Analysis results of 18-round LBlock for 2^{23} plaintext pairs

Θ	$\mathbf{P}(R \geq \Theta)$	$\mathbf{P}(W \geq \Theta)$	N_{wk}	Time Complexity
1.043	0.93	0.018	6.62e+14	$2^{74.25}$
2.245	0.87	0.007	2.75e+14	$2^{73.01}$
3.446	0.79	0.002	1.033e+14	$2^{71.67}$
4.647	0.69	0.0009	3.49e+13	$2^{70.31}$
6.449	0.5	0.0001	5.63e+12	$2^{68.71}$

5.2 Related-Key Attack on 20 and 21 Rounds

The related key truncated differential distribution attack applies to LBlock reduced to 20 and 21 rounds. Considering the key scheduling process, when the key difference goes through the S-boxes s_8 or s_9 the output difference is unknown. However, due to the slow avalanche effect of the key schedule, it takes multiple rounds for key differences to reach these S-boxes. Therefore, it is easy to find the truncated difference probability distribution for all the possible key differentials.



(a) 4-round SD phase (b) 10-round TDD phase (c) 6-round PKR phase

Fig. 6. Related-key truncated differential attack on 20-round LBlock

During the attack, we test each expected key differential in parallel to determine the correct key differential path.

The related key attack on 20 rounds consists of a 4-round SD phase, 10-round TDD, and 6-round PKR phase (see Fig. 6). The SD phase starts with 5 non-zero differences which are all cancelled through the 4 rounds differential trail, finishing with no difference in the output state. The key-register at the first round of the TDD phase has difference in just one bit (the 13th least significant bit). The key difference does not affect the round sub-keys for two rounds. The truncated differential distribution is calculated for the 10-round TDD phase. Nibble 5 (of the output right-hand half) has the highest KL-divergence $D(P||Q) = 2.189429e{-}03$ and is chosen as the target nibble. Finally, 6 final rounds are added as the PKR phase, requiring 52 key bits be guessed to reach the target nibble. From these key bits, two sub-key nibbles $SK[0]_2$ and $SK[0]_4$ are determined for the first round of the SD phase. Consequently, the input values of the active S-boxes are known in the first round and the overall probability of the SD phase increases to $\mathbf{P_{SD}} = 2^{-6}$.

Table 7, shows the results for the 20-round related key attack with different success rates. Note that the number of plaintext/ciphertext pairs includes the amount required to follow the SD phase. Considering the LLR threshold equal to the expected value of the right key ($E(R)$), with 2^{27} chosen plaintexts ($N = 2^{23}$ right-pairs), the complexity of the key recovery attack is $2^{74.55}$.

The related-key attack is extended to 21 rounds by adding one more round to the beginning of the SD phase in the above 20-round attack. Fig. 7 shows the SD phase in 21-round attack. The other phases are similar to the ones in the 20-round attack. If the attacker guesses 5 more key bits in the PKR phase (a total of 57 bits), he finds the 3 sub-key nibbles ($SK[0]_1$, $SK[0]_2$ and $SK[0]_4$) required to know the values of the active S-boxes in the first SD round. Also, the input values of 2 active S-boxes in the second round is clear by knowing sub-key nibbles $SK[0]_0$, $SK[0]_5$, $SK[1]_2$ and $SK[1]_4$. The analysis results of the attack on 21 rounds is shown in Table 7. Overall, the related-key attack on 21-round LBlock is possible with $N_p = 2^{30}$ chosen plaintexts ($N = 2^{20}$ right-pairs) and $2^{77.56}$ time complexity, when the attack success rate is 50%.

Table 7. Related-key analysis results on reduced LBlock

Specification	Θ	$\mathbf{P}(R \geq \Theta)$	$\mathbf{P}(W \geq \Theta)$	N_{wk}	Time Complexity
20 rounds,	1.6798	0.84	0.0183	8.28e+13	$2^{75.36}$
$N_p = 2^{27}$ pairs,	3.7397	0.63	0.0029	1.31e+13	$2^{74.66}$
$E(R) = 4.7696$	4.7696	0.5	0.0010	4.51e+12	$2^{74.55}$
21 rounds,	-0.1320	0.74	0.3355	4.83e+16	$2^{78.61}$
$N_p = 2^{30}$ pairs,	0.2320	0.63	0.2240	3.23e+16	$2^{78.11}$
$E(R) = 0.5962$	0.5962	0.5	0.1373	1.98e+16	$2^{77.56}$

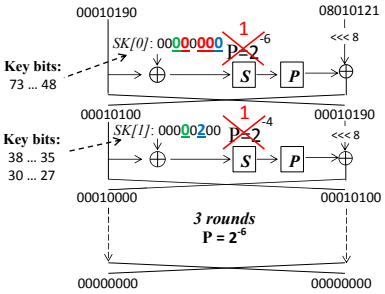


Fig. 7. The SD phase of the 21-round related-key attack

6 Conclusion

In this paper we presented truncated differential analysis of block cipher LBlock by analysing probability distribution of the truncated differences. Also we used LLR statistical test to employ the key-recovery attacks. The attack uses a distinguisher based on truncated differential distribution that are significantly different from a random permutation. Candidate sub-keys are guessed over several final rounds and the observed differences are measured against the expected distribution. We extend the distinguisher by concatenating additional rounds to the beginning which follow a classical differential characteristic. By exploiting the properties of the key schedule, we greatly increase the probabilities of differentials passing through the beginning rounds. We verified the analysis by implementing an example attack on 12-round LBlock and provide empirical data conforming the theory. Finally, we describe single-key and related-key attacks on LBlock reduced to 18 and 21 rounds, respectively. Finding probability distribution of the truncated differential, our attack can be applied on the ciphers with relatively large block size.

References

1. Albrecht, M.R., Leander, G.: An all-in-one approach to differential cryptanalysis for small block ciphers. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 1–15. Springer, Heidelberg (2013)
2. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 432–450. Springer, Heidelberg (2004)
3. Blondeau, C., Gérard, B., Nyberg, K.: Multiple differential cryptanalysis using `LLR` and χ^2 statistics. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 343–360. Springer, Heidelberg (2012)
4. Blondeau, C., Nyberg, K.: New links between differential and linear cryptanalysis. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 388–404. Springer, Heidelberg (2013)

5. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
6. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York (1991)
7. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional extension of Matsui's algorithm 2. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 209–227. Springer, Heidelberg (2009)
8. Hong, D., et al.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
9. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: A Block Cipher for IC-Printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)
10. Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)
11. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
12. Liu, S., Gong, Z., Wang, L.: Improved related-key differential attacks on reduced-round LBlock. In: Chim, T.W., Yuen, T.H. (eds.) ICICS 2012. LNCS, vol. 7618, pp. 58–69. Springer, Heidelberg (2012)
13. Liu, Y., Gu, D., Liu, Z., Li, W.: Impossible differential attacks on reduced-round LBlock. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 97–108. Springer, Heidelberg (2012)
14. Minier, M., Naya-Plasencia, M.: A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock. Information Processing Letters 112(16), 624–629 (2012)
15. Sasaki, Y., Wang, L.: Comprehensive study of integral analysis on 22-round LBlock. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 156–169. Springer, Heidelberg (2013)
16. Soleimany, H., Nyberg, K.: Zero-correlation linear cryptanalysis of reduced-round LBlock. In: International Workshop on Coding and Cryptography, WCC 2013 (2013)
17. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)