An Improved Truncated Differential Cryptanalysis of KLEIN

Shahram Rasoolzadeh[1,2], Zahra Ahmadian[1,2], Mahmoud Salmasizadeh[2], and Mohammad Reza Aref[1]

[1] Information Systems and Security Lab (ISSL), Department of Electrical Engineering,
[2] Electronic Research Institute,
Sharif University of Technology, Tehran, Iran
{sh_rasoolzadeh,ahmadian}@ee.sharif.edu,{salmasi,aref}@sharif.edu

Abstract. KLEIN is a family of lightweight block ciphers which proposed at RFIDSec 2011 by Gong et al. It has a 64-bit state and 64, 80 or 96-bit key size which introduce its version. It uses 16 same 4-bit Sboxes combined with two AES's MixColumn transformations for each round. This approach allows compact implementations of KLEIN in both low-end software and hardware. Such an innovative combination attracts the attention of cryptanalysts, and several security analyses have been published. The most successful one was represented in FSE'15 which was a truncated differential attack. They could attack up to 12, 13 and 14 rounds out of total number of 12, 16 and 20 rounds for KLEIN-64, -80 and -96, respectively. In this paper, by finding more efficient truncated differential paths and a slight improving in key recovery method we present two new truncated differential attacks on KLEIN, which recover the full secret key with better time and data complexities for the previously analyzed number of rounds. Also by using these truncated differential paths we are able to attack up to 14 and 15 rounds for KLEIN-80 and -96, respectively, which are the highest rounds ever analyzed.

Keywords: KLEIN, Truncated Differential Attack, Lightweight Block Cipher.

1 Introduction

Designig a secure and lightweight primitive for constrained environments such as RFID tags or wireless sensor networks is one of the interesting majors in cryptographic community. In order to find solutions for this ever-increasing demand, lightweight cryptography is developed as one of the most active areas in symmetric cryptography community. In this direction, a number of lightweight block ciphers have been proposed in the recent years, one of which is KLEIN block cipher [1].

KLEIN family of lightweight block ciphers is proposed by Gong et al. in RFIDSec 2011. It has three versions named KLEIN-64, -80 and -96, indicating the key size, with 12, 16 and 20 rounds respectively. It has a SPN structure,

Table 1. Summary of cryptanalytic results on KLEIN

Vrsion	Rounds	Time	Data	Memory	Attack Type	Ref.
KLEIN-64	7	$2^{45.5}$	$2^{34.3}$	2^{32}	Integral	[2]
	8	$2^{46.8}$	2^{32}	2^{16}	Truncated	[2]
	8	2^{35}	2^{35}	-	Truncated	[3]
	10	2^{62}	1	2^{60}	PC MitM*	[6]
	12	$2^{62.8}$	2^{39}	$2^{4.5}$	Biclique	[4]
	12	2^{57}	$2^{54.5}$	2^{16}	Truncated	[7]
	12	$\mathbf{2^{55.7}}$	$\mathbf{2^{48.6}}$	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
KLEIN-80	8	$2^{77.5}$	$2^{34.3}$	2^{32}	Integral	[2]
	11	2^{74}	2	2^{74}	PC MitM*	[6]
	13	2^{76}	2^{52}	2^{16}	Truncated	[7]
	14	$\mathbf{2^{75.9}}$	$\mathbf{2^{60.6}}$	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	14	$\mathbf{2^{78.9}}$	$\mathbf{2^{57.5}}$	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	16	2^{79}	2^{48}	2^{60}	Biclique	[5]
KLEIN-96	13	2^{94}	2	2^{82}	PC MitM*	[6]
	14	$2^{89.2}$	$2^{58.4}$	2^{16}	Truncated	[7]
	15	$\mathbf{2^{92.9}}$	$\mathbf{2^{63.5}}$	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	20	$2^{95.18}$	2^{32}	2^{60}	Biclique	[5]

* Parallel Cut Meet in the Middle

which combines 4-bit Sboxes with AES MixColumn. Such a combination allows a compact and low memory implementation in software and hardware, which results show that this cipher is utilizable in constrained-resource environments.

Despite some basic evaluations carried out on KLEIN by the designers [1], its real security level is not determined without further external analysis.So far, some cryptanalyses have been published on KLEIN, most of which exploiting the security drawbacks arisen from its innovative structure [2–7]. Apart from the biclique attacks [4, 5] which is inherently a bruteforce-like attack analysing the full round version, the most successful attack was discovered and exploited by Lallemand and Naya-Plasencia in FSE 2014 [7] which can recover the master key in full 12-round, reduced 13- and 14-round for KLEIN-64, -80 and -96, respectively.

In this paper, by finding new truncated differential paths and a slight improvement in key recovery method we present two new truncated differential attacks, which outperforms [7] in data and time complexities for full round KLEIN-64 and can analyze one round more for -80 and -92. The complexity of existing attacks and ours are summarized in Table 1.

This paper is organized as follows: Section 2 presents a brief description of KLEIN. In Section 3, new truncated differential paths, the outline of the key recovery attack on KLEIN with all details and its complexities evaluations are represented. Finally, Section 4 concludes this paper.

2 Description of KLEIN

KLEIN is a Substitution-Permutation Network (SPN) family of block ciphers with 64-bit block size and three types of key size that introduce its version: KLEIN-64, KLEIN-80 and KLEIN-96, which have 12, 16 and 20 rounds, respectively. Every round consists of four layers:

1. AddRoundKey (ARK): XORing the entering state with the round-key.
2. SubNibbles (SN): State is divided to 16 nibbles, and each nibble passed through a 4-bit Sbox.
3. RotateNibbles (RN): Rotating state two bytes to the left.
4. MixNibbles (MN): Applying AES's MixColumn transformation to each half of the state.

All 16 Sboxes are the same and the reason of this choice by designers is that a 4-bit Sbox has less implementation costs and memory compared to a 8-bit. Also for reducing the decryption costs, they choose an involutive Sbox[1].

An additional ARK layer is proceed after last round. So the encryption routine requires one more key than the number of rounds. The structure of one round of KLEIN is shown in Fig. 1. $X^{(r)}$ and $K^{(r)}$ are the input state and the subkey of round r, respectively.

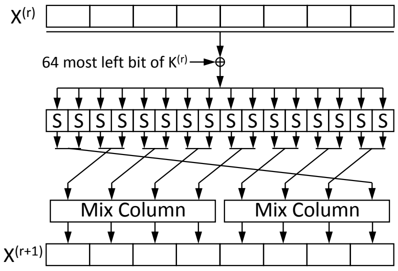

Fig. 1. Structure of one round of KLEIN

Let us focus on AES's MixColumn transformation, which works according to the following matrix multiplication in $GF(2^8)$ with the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$:

$$M = \begin{pmatrix} \mathsf{02\ 03\ 01\ 01} \\ \mathsf{01\ 02\ 03\ 01} \\ \mathsf{01\ 01\ 02\ 03} \\ \mathsf{03\ 01\ 01\ 02} \end{pmatrix} . \tag{1}$$

For reminding, multiplication of 02 by $x \in GF(2^8)$ can be performed as follows:

$$\mathsf{02} \times x = \begin{cases} x \ll 1 & \text{if MSB}(x) = 0 \\ x \ll 1 \oplus \mathsf{0x1b} & \text{if MSB}(x) = 1 \end{cases} , \tag{2}$$

where $x \ll n$ means shifting x, n bits to left and MSB is the most significant bit. Also the multiplication by 03 is equal to:

$$\mathsf{03} \times x = \mathsf{02} \times x \oplus x \tag{3}$$

These descriptions of finite field multiplications will be more useful in explaining the MN layer properties in the next section. It is better to note that only MN layer is byte-wise while the others can be seen as nibble-wise.

The KeySchedule of KLEIN is designed under implementation considerations. The round-keys are computed from the Master Key (MK) with the KeySchedule algorithm that follows a Feistel-like swap. The round keys $K^{(r)}$, $r = 1, \cdots, R$ (R is the number of rounds), and the final whitening key $K^{(R+1)}$ is generated as follows. First, the master key MK is stored in a key register as $K^{(1)}$. Then the following steps are iteratively applied to MK to generate R more subkeys:

1. Rotate the two halves of the key state to left one byte each.
2. Swap the two halves by a Feistel-like structure.
3. In left half of key state, xor 3^{rd} byte from left with round counter r.
4. In right half of key state, substitute 2^{nd} and 3^{rd} bytes using four KLEIN Sboxes.

At the end of round r, the leftmost 64 bit of the key register is $K^{(r+1)}$. Fig. 2 shows one round of the key schedule for KLEIN-64.

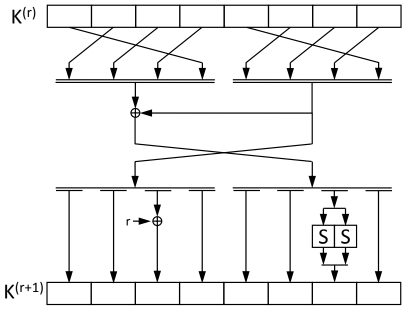

Fig. 2. Key schedule of one round of KLEIN-64

3 Truncated Differential Cryptanalysis of KLEIN

In this section, we will introduce two new truncated differential paths. Then we make use of key recovery method which was used in [7] first and for improving it we change it a little. Finally, the complexities of our attacks will be represented.

Proposition 1. [2,3] If the eight nibbles entering MixNibbles are of the form $0X0X0X0X$, where the wild-card X represents any 4-bit value, then the output is of the same form if and only if the MSB of the 4 lower nibbles all have the same value. This case occurs with probability 2^{-3}.

Proposition 2. If the eight nibbles entering MixNibbles are of the form $0X0X0X$ $0X$, then the output is the form of $00000X0X$ or $0X0X0000$ with probability of 31×2^{-15}.

Proposition 1 explained enough in previous cryptanalyses, especially in [7], so we don't discuss more about that. The proof of proposition 2 is as follow.

Proof. Consider $0A0B0C0D$ be the eight nibbles entering MixNibbles and $00000E$ $0F$ be the eight output nibbles. Also consider that $X = x_0x_1x_2x_3$, which x_0 is the MSB of X. As two most significant bytes of output is zero, we must have:

$$\begin{cases} B = 3 \times A \oplus 2 \times C \\ D = 7 \times A \oplus 7 \times C \end{cases} \Rightarrow \begin{cases} E = 11 \times A \oplus \ 9 \times C \\ F = 14 \times A \oplus 13 \times C \end{cases} \tag{4}$$

Since B, D, E and F are only four bits (higher nibbles in every byte are zero), it is equal to:

$$\begin{cases} c_0 = a_0 \\ c_1 = a_1 \\ c_2 = a_0 \oplus a_2 \end{cases} \tag{5}$$

Therefore, from $2^{16} - 1$ cases for A, B, C and D only 2^5 of them are acceptable. One of these 32 cases is all zeros which should be excluded. So the probability for this event is about 31×2^{-16}. By purposing second form of MixNibbles's output ($0E0F0000$) the probability would be 31×2^{-15}.

□

3.1 Truncated Differential Paths

Using proposition 1, an iterated truncated differential path for one round is presented in previous cryptanalyses [2,3], which has a probability of 2^{-6}. This iterated truncated differential path is shown in Fig. 3.

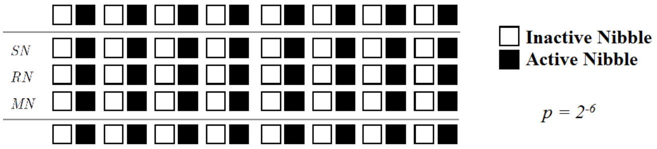

Fig. 3. Iterated truncated differential path for one round of KLEIN

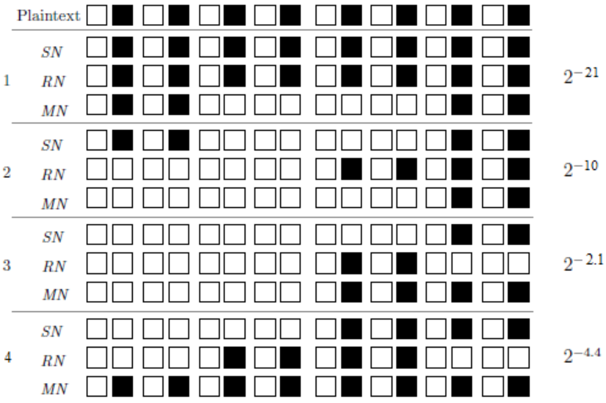

Fig. 4. Truncated differential path for 4 round of KLEIN

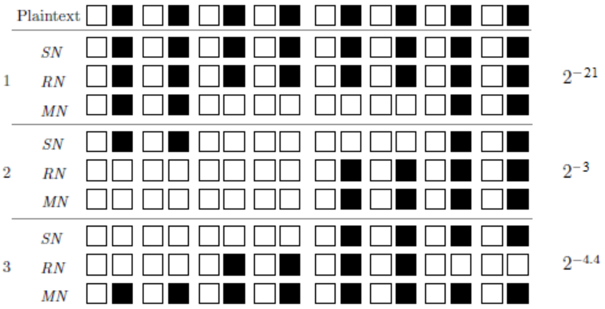

Fig. 5. Truncated differential path for 3 round of KLEIN

Also using propsition 2, we introduce two new truncated differential paths for four or three rounds that are shown in Fig. 4 and 5, respectively. In the first path we consider that the event which was introduced in propsition 2 happens for both of MixColumns of round 1 with a condition that output active nibbles be close to each other after RN layer. Then its probability is

$$p_1 = \frac{1}{2} \times (31 \times 2^{-15})^2 \simeq 2^{-21.1}. \tag{6}$$

Therefore only one MixColumn is active in round 2 and if the mentioned event happens again, its probability would be

$$p_2 = 31 \times 2^{-15} \simeq 2^{-10}. \tag{7}$$

So there are at most 2 active lower nibbles for input of the third round. These lower nibbles will activate only one MixColumn, and only lower nibbles in output of MixColumn will be active with probability of:

$$p_3 = \frac{2}{31} \times \frac{7}{15} + \frac{29}{31} \times (\frac{7}{15})^2 \simeq 2^{-2.1} \tag{8}$$

Regarding (8), it must be stated that for 2 cases of 31 cases, only one lower nibble is active, and when a nibble is active with probability one, the probability for that output difference of Sbox has a MSB equal to 0 is 7/15. After this input of each MixColumn in fourth round has at most 2 active lower nibbles. Probability of that output of fourth round have only active lower nibbles is:

$$p_4 = (\frac{7}{15})^4 \simeq 2^{-4.4} \tag{9}$$

The second path is look like the first one, except that event of the second round in first path is omitted. Therefore the probability for that only lower nibbles are activated is $p_2 = 2^{-3}$ for round 2 and $p_3 = 2^{-4.4}$ for round 3. In both of the paths, we will use introduced iterated truncated path for the reminding rounds. The probability for an $(R{-}1)$-round distinguisher of KLEIN will be $p = 2^{-6\times R-7.6}$ and $p = 2^{-6\times R-4.5}$, respectively using first and second path. As we will see, these two paths will be able to attack up to 14 and 15 rounds, respectively. It must be considered that in Fig. 4 or 5 only one side of the probability is shown.

3.2 Procedure of key recovery attack

For recovering key's lower nibbles we use a slightly modified version of the key recovery attack used in [7]. First we will bring two propositions that introduced in previous cryptanalyses. Using these propositions we will be able to partially decrypt the lower or higher nibbles in each round.

Proposition 3. [2,3] In the KeySchedule algorithm, lower nibbles and higher nibbles are not mixed: the lower/higher nibbles of any round-key can be computed directly from the lower/higher nibbles of the master key.

Proposition 4. [7] The values of the lower/higher nibbles outputing MN depend on the values of the lower/higher nibbles at the input and 3 more bits computed from the higher/lower nibbles that we will call the information bits. A similar property holds for the computation of the output lower/higher nibbles of MN^{-1}.

Proof of proposition 4 is given in [7] and we don't bring it here. These two propertise of KLEIN will let us to recover lower or higher nibbles of the master key. The key recovery method is as follow:

1. **Collecting enough pairs of data:** For being ensured to get one pair that conforms our differential path, we must generate a certain number of plaintexts. So we must have about p^{-1} pairs of data, but for reducing data complexity we use structural chosen plaintext attack. The size of structures will be determined by number of active bits in the truncated difference entering the first round. Then if our truncated differential have Δ active bits, the size

of structures will be 2^{Δ} plaintexts and every structure have about $2^{2\Delta-1}$ pairs.

For obtaining the required p^{-1} pairs, we must encrypt about $\frac{p^{-1}}{2^{\Delta-1}}$ plaintexts then this number is our data complexity. All 2^{Δ} plaintexts in a structure will be saved and processed and then be deleted, so we need a memory to save all these plaintexts. As we will see, this is our saleint memory complexity and other needs to memory is negligible.

2. **Sieving pairs of ciphertexts:** By inverting the ciphertext difference through the last MN we can observe the value of the difference entering this layer and then discard the ones that do not have the higher nibbles inactive. With this work we can eliminate such pairs that we are sure do not verify the differential path. Only 2^{-32} of wrong pairs can survive this filtering, so there will be $p^{-1} \times 2^{-32}$ remaining pairs of plaintexts.

   In practical, it is not necessary to invert all of ciphertext pairs, because if only lower nibbles in input of MN are active, the output higher nibbles could be only $0x0$ or $0x1$. Using this property, we can do ciphertext pairs sieving with a negligible time complexity.

3. **Guessing lower nibbles of first subkey:** For each remaining pair that has passed the filtering of the previous step, we will find possible values of the first 8 lower nibbles of the key in two levels. For event described in proposition 2 there are 2×31 possible input differences for both MixColumn, so 62×2^{16} pairs are possible for half of the output SN. Therefore there are normally 62 pairs which have the same differences. By passing these pairs from $\mathsf{SN}^{-1} = \mathsf{SN}$ and saving them and their differences in a table with sorting on their differences. So we can find all 62 possible keys for 4 lower nibbles only by xoring the plaintexts with pairs of table that difference of pairs are equal to the first 4 nibbles in plaintexts difference.

   Using this method again we can find 31 possible keys for other 4 lower nibbles. In other meaning, for each pair of plaintexts and their ciphertexts that pass the previous step, we have 2×31^2 key candidate for the 8 first lower nibbles of the master key.

   This step requires a negligible time complexity because all used operations are xoring, and this let us to compute half of both states at the input of the first MN that already satisfies the conditions of round 1. As in [7], this pair of half states will be denoted by $(S, S')^*$.

   For KLEIN-64, the lower nibbles of first subkey determine all the lower nibbles of the whole key but for obtaining all the possible lower nibble values of KLEIN-80 and KLEIN-96, we have to make additional guesses for other 8 and 16 bits of lower nibbles, respectively. After this step, we will have $p^{-1} \times 2^{-32} \times 2 \times 31^2$, $p^{-1} \times 2^{-32} \times 2 \times 31^2 \times 2^8$ and $p^{-1} \times 2^{-32} \times 2 \times 31^2 \times 2^{16}$ possible candidates (C, C', k_{low}), respectively for KLEIN-64, KLEIN-80 and KLEIN-96.

4. **Sieving candidate subkeys on second round:** For first path in round 2, the mentioned event of proposition 2 happens again. We will use the saved table again to know whether candidates for the lower nibbles of key can pass this round or not. Because the values of active nibbles after SN of first round in both plaintexts are known, we can guess values of active lower nibbles after MN layer (We will path value of four nibbles through MixColumn. This value is one of the possible value and the other one is this value xored with 0xb). So we will search through 2^4 possible differences in the table to examine if that value of 4 nibbles xored with corresponding 4 nibbles of subkey of candidate key is equal with the plaintext values saved in table or not. A plaintext pair and a candidate key can pass through this sieve with probability of $62 \times 2^{-16} \times 2^4$. Note that this step will be used only with the first path. Like previous step, time complexity of this step is negligible.

5. **Inverting pairs of ciphertexts:** At this step we will invert every possible triple of (C, C', k_{low}), generating possible pairs $(S, S')_r$ for $r = R, R-1, \ldots,$ (which $(S, S')_r$ shows the value of lower niblles entering round $r+1$). As for every MixColumn we have 3 information bits, inverting one round costs 2×2^3 round encryptions per triple. During the iterative rounds, the number of possible triples stays the same, because every 2^6 state of inverting only 2^{-6} of them can satisfy the condition of proposition 1. But during the non-iterative rounds, because of tight conditions, number of candidates gets reduced.
   Once we have computed $(S, S')_2$ (for the first path and $(S, S')_1$ for the second path), we have to guess the 3 bits needed to invert the second (for the first path and first for the second path) MN, and then we have to match values and active differences with the already computed values $(S, S')^*$. After the matching condition, number of key candidates for a pair of ciphertexts gets so smaller than $2^{k_{low}}$. So, the cost of recovering the key is much smaller than an exhaustive search.
   The cost of this step is given by the number of candidate triples multiplied by 2^4 (cost of inverting one round), multiplied by the number of iterative rounds. The cost for inverting non-iterative rounds are so small, because the number of candidates have been reduced so much. Time complexity for the other steps are negligable, this step will determine this attack's time complexity.

6. **Recovering higher nibbles of master key:** If a k_{low} candidate for a pair of ciphertext and their corresponding plaintext can path matching condition in previous step, with an exhaustive search for higher nibbles we will find the whole bits of the master key.

3.3 Results and Complexities

Applying described key recovery attack to paths 1 and 2, we will be able to attack up to 14 and 15 rounds KLEIN which cases introduced in [7] could not reach. Results of our attacks are shown in Table 3. In all of our attacks the memory complexity is 2^{32} of 64-bit words (size of plaintext).

As it can be seen, using the first path makes a good time complexity and the second path a good data complexity. These have a trade-off between time and data. Our attack to full-round KLEIN-64 and reduced 13-round KLEIN-80 can retrieve full master key with 2.5 times faster and $\frac{1}{64}$ of data of [7]. About the attack to reduced 14-round KLEIN-96, our attack by using the first path is 40 times but data complexity is 2 times more than that of [7], and using the second path, it is 5 times faster while it requires half of data complexity to that of [7]. Except biclique attacks, cryptanalyzing reduced 14-round KLEIN-80 and reduced 15-round KLEIN-96 are introduced for first time.

Table 2. Summary of the complexities of our attacks

Vrsion/Rounds	Path	Probability	Time	Data
KLEIN-64/12	I	$2^{-79.6}$	$2^{55.7}$	$2^{48.6}$
	II	$2^{-76.5}$	$2^{58.8}$	$2^{45.5}$
KLEIN-80/13	I	$2^{-85.6}$	$2^{69.8}$	$2^{54.6}$
	II	$2^{-82.5}$	$2^{72.9}$	$2^{51.5}$
KLEIN-80/14	I	$2^{-91.6}$	$2^{75.9}$	$2^{60.6}$
	II	$2^{-88.5}$	$2^{78.9}$	$2^{57.5}$
KLEIN-96/14	I	$2^{-91.6}$	$2^{83.9}$	$2^{60.6}$
	II	$2^{-88.5}$	$2^{86.9}$	$2^{57.5}$
KLEIN-96/15	II	$2^{-94.5}$	$2^{92.9}$	$2^{63.5}$

4 Conclusions

In this paper we introduced two new truncated differential paths for KLEIN, as well as an improved key recovery method based on what proposed by Lallemand and Naya-Plasencia. Results show that our attacks have the best time and data complexities on full-round KLEIN-64, reduced 13-round KLEIN-80 and reduced 14-round KLEIN-96 so far. Also, we introduced two new attacks on reduced 14-round KLEIN-80 and reduced 15-round KLEIN-96 for first time.

The block cipher KLEIN has two main weaknesses: 1. MixNibbles layer using Rijndael's MixColumn transformation does not correctly mix higher and lower nibbles, as it is the only transform that does. 2. KeySchedule does not mix higher and lower nibbles. These two helps the cryptanalyst to perform a reduced partial key search, so maybe considering other diffusion layers instead of Rijndael's and a stronger KeySchedule could help to prevent the attacks.

References

1. Z. Gong, S. Nikova, and Y. W. Law. “KLEIN: A new family of lightweight block ciphers”. In RFIDSec, LNCS, vol. 7055, pp. 1-18. Springer, 2012.
2. X. Yu, W. Wu, Y. Li, and L. Zhang. “Cryptanalysis of reduced-round KLEIN block cipher”. In Inscrypt, LNCS, vol. 7537, pp. 237-250. Springer, 2012.
3. J. P. Aumasson, M. Naya-Plasencia, and M. J. O. Saarinen. “Practical attack on 8 rounds of the lightweight block cipher KLEIN”. In INDOCRYPT, LNCS, vol. 7107, pp. 134-145. Springer, 2011.
4. Z. Ahmadian, M. Salmasizadeh, and M. R. Aref. “Biclique Cryptanalysis of the Full-Round KLEIN Block Cipher”. Accepted in IET Information Security, 2014.
5. F. Abed, C. Forler, E. List, S. Lucks, and J. Wenzel. “Biclique Cryptanalysis Of PRESENT, LED, And KLEIN”. Cryptology ePrint Archive, Report 2012/591, 2012. http://eprint.iacr.org/
6. Ivica Nikolic, Lei Wang, and Shuang Wu. “The Parallel-Cut Meet-In-The-Middle Attack”. Cryptology ePrint Archive, Report 2013/530, 2013. http://eprint.iacr.org/
7. V. Lallemand and M. Naya-Plasencia.“Cryptanalysis of KLEIN”. In FSE 2014 accepted papers, full version available at Cryptology ePrint Archive, Report 2014/090, 2014. http://eprint.iacr.org/