An Improved Truncated Differential Cryptanalysis of KLEIN

Shahram Rasoolzadeh[1], Zahra Ahmadian[2], Mahmoud Salmasizadeh[3], and Mohammad Reza Aref[3]

[1] Simula Research Laboratory, Bergen, Norway,
[2] Shahid Beheshti University, Tehran, Iran,
[3] Sharif Technical University, Tehran, Iran.

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 2014 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 two new truncated differential paths with better probabilities and a slight changing in key recovery method we present two 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, block cipher, lightweight.

1 Introduction

Designing 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 an SPN structure, which combines 4-bit Sboxes with AES's 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 of 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 brute-force-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-, reduced 13- and 14-round for KLEIN-64, -80 and -96, respectively.

Truncated differential attack is a generalization of differential attack, that Lars Knudsen developed the technique in 1994 [8]. Whereas ordinary differential cryptanalysis analyzes the full difference between two texts, the truncated variant considers differences that are only partially determined. That is, the attack makes predictions of only some of the bits instead of the full state.

In this paper, by finding new truncated differential paths and a slight changing in key recovery method we present two truncated differential attacks, which outperform [7] in data and time complexities for full round KLEIN-64, 13-round KLEIN-80 and 14-round KLEIN-96. Also these attacks can analyze one round

Table 1. Summary of cryptanalytic results on KLEIN

Version	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	$\mathbf{2^{54.9}}$	$\mathbf{2^{48.6}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	12	$\mathbf{2^{58.0}}$	$\mathbf{2^{45.5}}$	$\mathbf{2^{32}}$	**Truncated**	**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]
	13	$\mathbf{2^{69.0}}$	$\mathbf{2^{54.6}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	13	$\mathbf{2^{72.0}}$	$\mathbf{2^{51.5}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	14	$\mathbf{2^{75.0}}$	$\mathbf{2^{60.6}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	14	$\mathbf{2^{78.0}}$	$\mathbf{2^{57.5}}$	$\mathbf{2^{32}}$	**Truncated**	**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]
	14	$\mathbf{2^{83.0}}$	$\mathbf{2^{60.6}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	14	$\mathbf{2^{86.1}}$	$\mathbf{2^{57.5}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	15	$\mathbf{2^{92.1}}$	$\mathbf{2^{63.5}}$	$\mathbf{2^{32}}$	**Truncated**	**4**
	20	$2^{95.18}$	2^{32}	2^{60}	Biclique	[5]

* Parallel Cut Meet in the Middle

more for KLEIN-80 and KLEIN-96. 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 will be introduced and in Section 4 the outline of the key recovery attack on KLEIN with all details and its complexities evaluations are presented. Finally, Section 5 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): *Xor*-ing 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 Figure 1 that $X^{(r)}$ and $K^{(r)}$ are the input state and the subkey of round r, respectively.

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

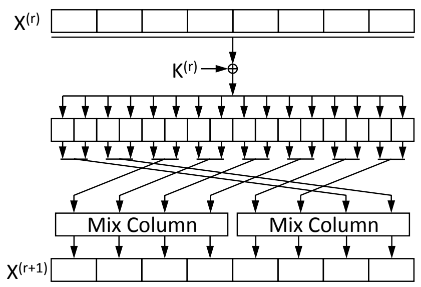

Fig. 1. Round structure of KLEIN

$x^8 + x^4 + x^3 + x + 1$:

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

For reminding, multiplication of 2 in this transformation can be performed as follows:

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

where $x \ll n$ means shifting x, n bits to left and msb is the most significant bit. Also the multiplication by 3 is equal to:

$$\mathsf{3} \times x = \mathsf{2} \times x \oplus x \tag{3}$$

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 Key Schedule of KLEIN is designed under implementation considerations. The round keys are computed from the master key with the Key Schedule 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 is stored in a key register as $K^{(1)}$. Then the following steps are iteratively applied generate R more subkeys:

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

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

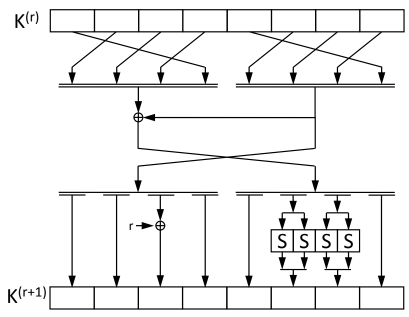

Fig. 2. Key schedule of one round of KLEIN-64

3 Truncated Differential Paths

In this section, we will introduce two new truncated differential paths which have better probabilities than the truncated differential paths in [7].

Proposition 1. [2, 3] If the eight nibbles entering MixColumn 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 MixColumn are of the form $0X0X$ $0X0X$, 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 do not discuss more about that. The proof of *Proposition 2* is as follow.

Proof. Consider $0A0B0C0D$ be the eight nibbles entering MixColumn and 0000 $0E0F$ 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 31×2^{-16}. By purposing second form of MixColumn's output ($0E0F0000$) the probability would be 31×2^{-15}.

$\square$

Using *Proposition 1*, an iterated truncated differential path for one round is presented in previous cryptanalyses [2, 3]. Its probability is 2^{-6} which is caused by using the event in the *Proposition 1* twice. This iterated truncated differential path is shown in Figure 3.

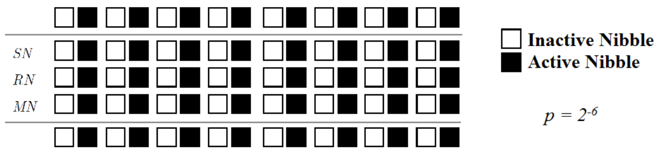

Fig. 3. Iterated truncated differential path for one round of KLEIN

Also using *Proposition 2*, we introduce two new truncated differential paths for four or three rounds that are shown in Figures 4 and 5, respectively. In the first path (path I), we consider that the event which was introduced in *Proposition 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 (path II) is look like the first one, except that event of the second round in first path is omitted. Therefore the probability for that only

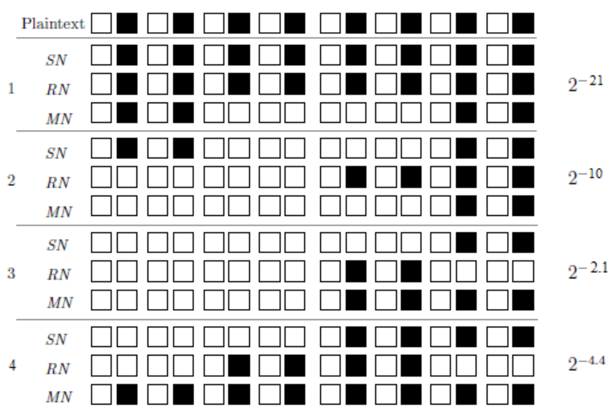

Fig. 4. Truncated differential path for 4 round of KLEIN

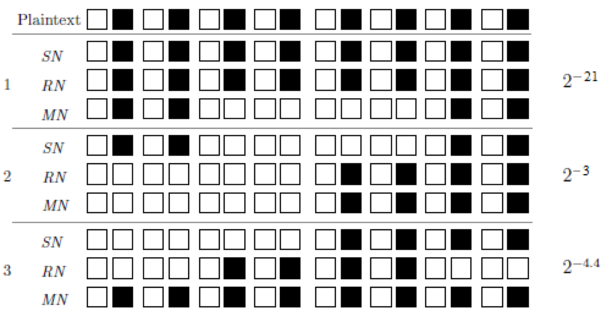

Fig. 5. Truncated differential path for 3 round of KLEIN

lower nibbles are activated is $p_2 = 2^{-3}$ for second round and $p_3 = 2^{-4.4}$ for third round. 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 paths I and II, respectively. As we will see, using these two paths, we will be able to attack up to 14 and 15 rounds, respectively. It must be considered that in Figure 4 or 5 only one side of the probability is shown.

4 Truncated Differential Cryptanalysis of KLEIN

In this section, we will make use of key recovery method which was used in [7] first and we improve it a little to fix it to the truncated differential paths introduced in previous section. At he last, the complexities of our attacks will be represented.

For recovering master 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 Key Schedule 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 outputting MixColumn 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 them information bits. A similar property holds for the computation of the output lower/higher nibbles of inverse MixColumn.

Proof of *Proposition 4* is given in [7] and we don't bring it here. These two properties of KLEIN will let us to recover lower and then 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} differential pairs, and for reducing data complexity we use structural chosen plaintext attack. The size of each structure will be determined by number of active bits in the truncated difference entering the first round. As our truncated differential paths have 32 active bits in the plaintext, the size of structures will be 2^{32} plaintexts and each structure has about $2^{2\times 32-1} = 2^{63}$ pairs.
   For obtaining the required p^{-1} differential pairs, we must encrypt about $p^{-1} \times \frac{2^{32}}{2^{63}} = p^{-1} \times 2^{-31}$ plaintexts then this number is our data complexity. All 2^{32} 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 salient memory complexity and other needs to memory is negligible.
2. **Sieving ciphertext pairs:** As in a right differential pair, the differential in the output of round $R-1$ has differences just in lower nibbles and this difference will continue until the input of MN layer. By inverting the ciphertext difference through the last MN layer, we can observe the value of the difference entering this layer and then discard the ones that have the active higher nibbles. With this work we can eliminate such pairs that we are sure do not verify the differential path (wrong differential pairs). Only fraction of 2^{-32} wrong pairs can survive this filtering, so there will be $p^{-1} \times 2^{-32}$ remaining differential pairs.
   In practical, it is not necessary to invert all of ciphertext pairs, because if only lower nibbles in input of MN layer are active, the output higher nibbles could be only 0 or 1. Using this property, we can sieve ciphertext pairs with a negligible time complexity.
3. **Guessing lower nibbles of first subkey:** For each remaining differential pair that has passed the sieving 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 each MixColumn, so 62×2^{16} pairs are possible for half of the output SN. Therefore there are normally 62 pairs which have the same difference in the input of SN. By passing these pairs from $\mathsf{SN}^{-1} = \mathsf{SN}$ and saving the input pair to SN and their output difference before MixColumn in a table sorted by their difference in the input of SN, we can find all 62 possible keys for 4 lower nibbles only by xoring the plaintexts with pairs in the table that difference of pairs are equal to the corresponding 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 there are only two look-ups to precomputed table and all other used operations are xoring. This lets us to compute half of both states at the input of the first MN layer that already satisfies the conditions of round 1. 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 $p^{-1} \times 2^{-31} \times 31^2 \times 2^{8\times(0,1,2)}$ possible candidates of (C, C', k_{low}), respectively for KLEIN-64, -80 and -96.

4. **Sieving candidate subkeys on second round:** In second round of path I, the mentioned event of *Proposition 2* happens again. We can 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 input lower nibbles for one of MixColumns in first round for both plaintexts are known, we can guess their values after MixColumn (We will path value of four nibbles through MixColumn. This value is one of the possible values and the other one is this value xored with `0xb`). But for each of 2^4 possible pairs the difference is the same. So we will search through this difference 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 saved values in table or not.

   A plaintext pair and a candidate key can pass through this sieve with probability of $62 \times 2^{-16} \times 2^4$, so there will be $p^{-1} \times 2^{-42} \times 31^3 \times 2^{8\times(0,1,2)}$ possible candidates (C, C', k_{low}), respectively for KLEIN-64, -80 and -96. Note that this step will be used only with the path I. 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 nibbles entering round r).

   As for every MixColumn we have 3 information bits, inverting one round costs 2^3 round encryptions per triple. During the iterative rounds, the number of possible triples stays the same, because from 2^6 possible values 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 significantly (factor of reduction for each event of *Proposition 2* is 2^{-11}).

   Once we have computed $(S, S')_3$ (for the first path and $(S, S')_2$ for the second path), we have to guess the 3 bits needed to invert the second (or first) one MixColumn, and then we have to match values 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 2^3 (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 negligible, this step will determine time complexity for this attack.

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

4.1 Results and Complexities

Applying described key recovery attack to both paths, 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 block size.

As it can be seen, using path I makes a good time complexity and path II makes a good data complexity. These have a trade-off between time and data. For the attacks with same number of rounds, despite of more memory complexity, our attacks are faster in time or need less data than attacks in [7] (and in some cases both). Also, except biclique attacks, cryptanalyzing reduced 14-round KLEIN-80 and reduced 15-round KLEIN-96 are introduced for first time.

5 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. Key Schedule does not mix higher and lower nibbles. These two help the cryptanalyst to perform a reduced partial key search, so maybe considering other diffusion layer instead of Rijndael's and a stronger Key Schedule could help to prevent the attacks.

Table 2. Summary of the complexities of our attacks

Version/Rounds	Path	Probability	Time	Data
KLEIN-64/12	I	$2^{-79.63}$	$2^{54.91}$	$2^{48.63}$
	II	$2^{-76.49}$	$2^{57.98}$	$2^{45.49}$
KLEIN-80/13	I	$2^{-85.63}$	$2^{68.96}$	$2^{54.63}$
	II	$2^{-82.49}$	$2^{72.02}$	$2^{51.49}$
KLEIN-80/14	I	$2^{-91.63}$	$2^{75.01}$	$2^{60.63}$
	II	$2^{-88.49}$	$2^{78.05}$	$2^{57.49}$
KLEIN-96/14	I	$2^{-91.63}$	$2^{83.01}$	$2^{60.63}$
	II	$2^{-88.49}$	$2^{86.05}$	$2^{57.49}$
KLEIN-96/15	II	$2^{-94.49}$	$2^{92.08}$	$2^{63.49}$

References

1. Z. Gong, S. Nikova, and Y. W. Law. *"KLEIN: A new family of lightweight block ciphers"*. 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"*. 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"*. 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"*. Information Security Journal, vol. 9, issue 5, pp. 294-301. IET, 2015.
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.
6. I. Nikolic, L. Wang, and Sh. Wu. *"The Parallel-Cut Meet-In-The-Middle Attack"*. Cryptology ePrint Archive, Report 2013/530, 2013.
7. V. Lallemand and M. Naya-Plasencia. *"Cryptanalysis of KLEIN"*. FSE 2014, LNCS, vol. 8540, pp. 451-470. Springer, 2015
8. L. R. Knudsen. *"Truncated and Higher Order Differentials "*. FSE 1994, LNCS, vol. 1008, pp. 196-211. Springer, 1994.