DE GRUYTER
OPEN

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2016-0036
Tatra Mt. Math. Publ. **67** (2016), 135–147

AN IMPROVED TRUNCATED DIFFERENTIAL CRYPTANALYSIS OF KLEIN

Shahram Rasoolzadeh — Zahra Ahmadian –
– Mahmoud Salmasizadeh — Mohammad Reza Aref

ABSTRACT. KLEIN is a family of lightweight block ciphers which was proposed at RFIDSec 2011 by Gong et. al. It has three versions with 64, 80 or 96-bit key size, all with a 64-bit state size. It uses 16 identical 4-bit S-boxes 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 unconventional combination attracts the attention of cryptanalysts, and several security analyses have been published. The most successful one was presented at 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, we present improved attacks on three versions of KLEIN block cipher, which recover the full secret key with better time and data complexities for the previously analyzed number of rounds. The improvements also enable us to attack up to 14 and 15 rounds for KLEIN-80 and -96, respectively, which are the highest rounds ever analyzed. Our improvements are twofold: the first, finding two new truncated differential paths with probabilities better than that of the previous ones, and the second, a slight modification in the key recovery method which makes it faster.

1. Introduction

Designing a secure and lightweight primitive for constrained environments such as RFID tags or wireless sensor networks is one of the interesting trends 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 the KLEIN block cipher [1].

KLEIN family of lightweight block ciphers was proposed by Gong et al. in RFIDSec 2011. It has three versions named KLEIN-64, -80 and -96, indicating

2010 Mathematics Subject Classification: 94A60, 68P25.
Keywords: KLEIN, truncated differential attack, block cipher, lightweight.

the key size, with 12, 16 and 20 rounds respectively. It has an SPN structure, which combines 4-bit S-boxes with AES's MixColumn. Such a combination allows a compact and low memory implementation in software and hardware, making KLEIN a utilizable block cipher in constrained-resource environments.

Despite some basic evaluations carried out on KLEIN by the designers [1], its real security level cannot be determined without further external analysis. So far, some cryptanalyses have been published on KLEIN, most of which exploit the security drawbacks arisen from its unconventional 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 L a l l e m a n d and N a y a- P l a s e n c i a 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 the differential attack developed by K n u d s e n 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 about only some of the bits instead of the full state.

In this paper, by exploiting new truncated differential paths and a slight modification in key recovery method, we present new 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, for the first time, we propose cryptanalysis for 14 and 15 rounds of 80- and 96-bit versions of KLEIN, respectively. The complexities of existing attacks along with 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 are introduced and in Section 4 the outline of the key recovery attack on KLEIN with all details and its complexity evaluations are presented. Finally, Section 5 concludes this paper.

2. Description of KLEIN

KLEIN is a family of block ciphers with three variants KLEIN-64, KLEIN-80 and KLEIN-96 which have 64, 80 and 96 bits key, respectively. It is a Substitution-Permutation Network (SPN) with 64-bit block size for all versions, and 12, 16 and 20 rounds for KLEIN-64, KLEIN-80 and KLEIN-96, 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 into 16 nibbles, and each nibble is passed through a 4-bit S-box.

(3) RotateNibbles (RN): Rotating the state two bytes to the left.

(4) MixNibbles (MN): Applying AES's MixColumn transformation to each 32-bit half of the state.

All 16 S-boxes are the same and the reason for this choice by designers is that a 4-bit S-box has smaller implementation and memory costs compared to an 8-bit one. Also for reducing the decryption costs, they chose an involutive S-box [1].

TABLE 1. Summary of cryptanalytic results on KLEIN.

Version	Rounds	Time	Data		Memory	Attack Type	Ref.
KLEIN-64	7	$2^{45.5}$	$2^{34.3}$	CP	2^{32}	Integral	[2]
	8	$2^{46.8}$	2^{32}	CP	2^{16}	Truncated	[2]
	8	2^{35}	2^{35}	CP	–	Truncated	[3]
	10	2^{62}	1	KP	2^{60}	PC MitM*	[6]
	12	$2^{62.8}$	2^{39}	CP	$2^{4.5}$	Biclique	[4]
	12	2^{57}	$2^{54.5}$	CP	2^{16}	Truncated	[7]
	12	$\mathbf{2^{54.9}}$	$\mathbf{2^{48.6}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	12	$\mathbf{2^{58}}$	$\mathbf{2^{45.5}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
KLEIN-80	8	$2^{77.5}$	$2^{34.3}$	CP	2^{32}	Integral	[2]
	11	2^{74}	2	KP	2^{74}	PC MitM*	[6]
	13	2^{76}	2^{52}	CP	2^{16}	Truncated	[7]
	13	$\mathbf{2^{69}}$	$\mathbf{2^{54.6}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	13	$\mathbf{2^{72}}$	$\mathbf{2^{51.5}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	14	$\mathbf{2^{75}}$	$\mathbf{2^{60.6}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	14	$\mathbf{2^{78}}$	$\mathbf{2^{57.5}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	16	2^{79}	2^{48}	CP	2^{60}	Biclique	[5]
KLEIN-96	13	2^{94}	2	KP	2^{82}	PC MitM*	[6]
	14	$2^{89.2}$	$2^{58.4}$	CP	2^{16}	Truncated	[7]
	14	$\mathbf{2^{83}}$	$\mathbf{2^{60.6}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	14	$\mathbf{2^{86.1}}$	$\mathbf{2^{57.5}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	15	$\mathbf{2^{92.1}}$	$\mathbf{2^{63.5}}$	CP	$\mathbf{2^{32}}$	**Truncated**	**Sec. 4**
	20	$2^{95.2}$	2^{32}	CP	2^{60}	Biclique	[5]

* Parallel Cut Meet in the Middle KP/CP: Known/Chosen Plaintext

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

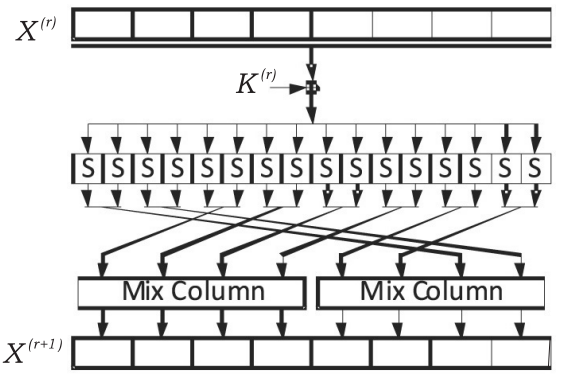

FIGURE 1. Round structure of KLEIN.

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

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

Recall that multiplying by 2 in this transformation can be performed as follows:

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

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

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

These descriptions of finite field multiplications are more useful for 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 generates $R + 1$ round keys $K^{(r)}$, $r = 1, \ldots$ $\ldots, R + 1$ from the master key, where R is the number of rounds of the cipher. The final subkey $K^{(R+1)}$ is used as the whitening key. KLEIN's key schedule works as follows. First, the master key is stored in a key register as $K^{(1)}$.

Then the following steps are iteratively applied to 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 the left half of the key state, xor the 3rd byte from the left with round counter r.
(4) In the right half of key state, substitute the 2nd and the 3rd bytes using four KLEIN S-boxes.

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.

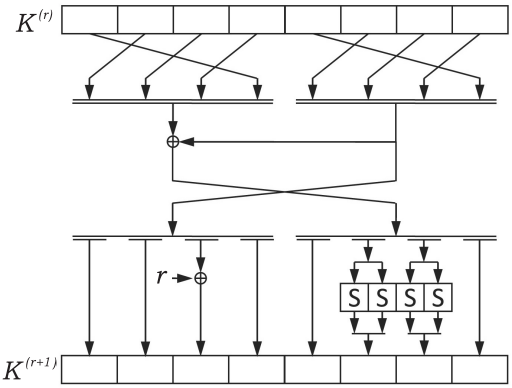

FIGURE 2. Key schedule of one round of KLEIN-64.

3. Truncated differential paths

In this section, we will introduce two new truncated differential paths with better probabilities than that of [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 all the four lower nibbles are the same. This case occurs with probability* 2^{-3}*.*

PROPOSITION 2. *If the eight nibbles entering MixColumn are of the form* $0X0X$ $0X0X$*, then the output will be of the form* $00000X0X$ *or* $0X0X0000$ *with probability of* 31×2^{-15}*.*

Proposition 1 has been explained enough in previous cryptanalyses, especially in [7], so we do not discuss it here. The proof of Proposition 2 is as follow.

P r o o f. Consider $0A0B0C0D$ be the eight nibbles entering MixColumn and $0000\ 0E0F$ be the eight output nibbles. Also consider that $X = x_0x_1x_2x_3$, where x_0 is the MSB of X. Since two most significant bytes of output are 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 have only 4 nonzero bits (e.g., the higher nibble in each byte is zero), it holds that

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

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 of this event is 31×2^{-16}. By taking the second form of MixColumn's output ($0E0F0000$) into account, the probability would be 31×2^{-15}. □

Using Proposition 1, an iterated truncated differential path for one round of KLEIN has been presented in [2], [3]. Its probability is 2^{-6} assuming the intersection of two independent events explained in Proposition 1. This iterated truncated differential path is shown in Fig. 3.

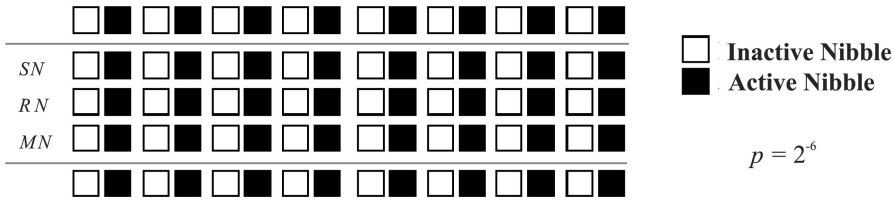

FIGURE 3. Iterated truncated differential path for one round of KLEIN.

Also using *Proposition 2*, we introduce two new truncated differential paths for four and three rounds that are shown in Fig. 4 and 5, respectively. In the first path (path I), we consider that the event explained in *Proposition 2* happens for both of MixColumns of round 1 and all output active nibbles belong to a single half state, after RN layer of round 2. Then its probability is

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

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

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

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

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

Regarding p_3, it must be stated that for 2 cases out of 31 cases, only one lower nibble is active, and when a nibble is active with probability one, the probability that the output difference of S-box has a MSB equal to 0 is $\frac{7}{15}$. After this, the input of each MixColumn in the fourth round has at most 2 active lower nibbles. The probability that only the lower nibbles of the fourth round's output are active is

$$p_4 = \left(\frac{7}{15}\right)^4 \simeq 2^{-4.4}.$$

The second path (path II) looks like the first one, except that the events of the second round in first path are omitted. Therefore the probability that only lower nibbles are activated is $p_2 = 2^{-3}$ for the second round and $p_3 = 2^{-4.4}$ for the third round. In both of the paths, we use the above mentioned iterated truncated path for the remaining rounds. The probability for an $(R-1)$-round distinguisher of KLEIN is $p = 2^{-6\times R-7.6}$ and $p = 2^{-6\times R-4.5}$, 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 Fig. 4 or Fig. 5 only one side of the probability is shown.

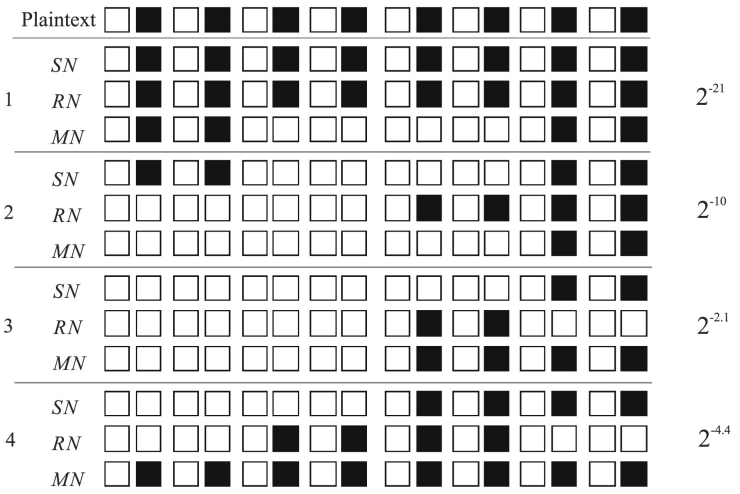

FIGURE 4. Truncated differential path for 4 round of KLEIN.

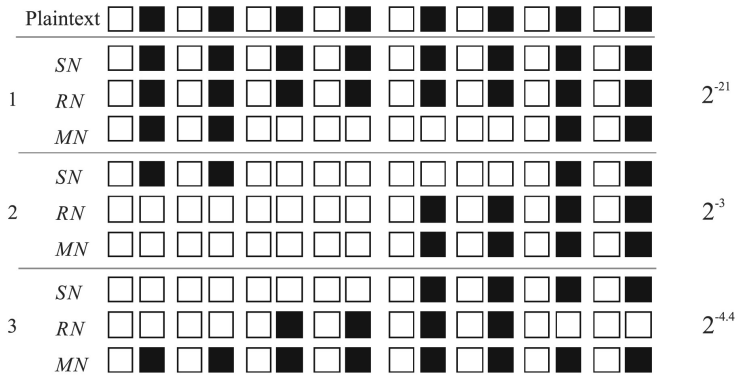

FIGURE 5. Truncated differential path for 3 round of KLEIN.

4. Truncated differential cryptanalysis of KLEIN

In this section, we first make use of the key recovery method presented in [7] and then we improve it slightly to proportionate it to the truncated differential paths introduced in the previous section. Finally, the complexities of our attacks will be presented.

For recovering master key's lower nibbles we use a slightly modified version of the key recovery attack used in [7]. First, we will state two propositions that were 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 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 do not state it here. These two properties of KLEIN will allow us to recover lower and then higher nibbles of the master key. The key recovery method is as follows:

(1) **Collecting enough pairs of data:** To be sure that we get one pair that conforms to 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 is determined by the 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 would be 2^{32} plaintexts and each structure has about $2^{2\times 32-1} = 2^{63}$ pairs.

To obtain the required p^{-1} differential pairs, we must encrypt about $p^{-1} \times \frac{2^{32}}{2^{63}} = p^{-1} \times 2^{-31}$ plaintexts. So, this value would be the attack data complexity. All 2^{32} plaintexts in a structure should be saved and processed, and then the allocated memory should be released. Therefore, we need memory space to save all these plaintexts. As we will see, this is our salient memory complexity and the memory required for other parts of the attack is negligible compared to this one.

(2) **Sieving ciphertext pairs:** As in the right differential pair, the differential in the output of round $R-1$ has differences just in lower nibbles and this difference will be preserved 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. In this way, we can eliminate such pairs that we are sure do not satisfy the differential path (wrong differential pairs). Only a fraction of 2^{-32} wrong pairs can survive this filtering, so there would be $p^{-1} \times 2^{-32}$ remaining differential pairs.

In practice, it is not necessary to invert all 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 the first subkey:** For each remaining differential pair that has passed the sieving of the previous step, we will find possible values for 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, a number of 62×2^{16} pairs is possible for half of the outputs of SN. Therefore, there are on average 62 pairs which have the same difference in the input of SN, on average. By passing these pairs through $\mathsf{SN}^{-1} = \mathsf{SN}$ and saving the input pair to SN and their output difference before MixColumn in a table indexed by the input of SN difference, we can find all 62 possible keys for 4 lower nibbles only by xoring the plaintexts with pairs in the table, where the difference of pairs is equal to the corresponding 4 nibbles in plaintexts difference.

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

This step requires a negligible time complexity because there are only two look-up tables and all the other used operations are XORs. This enables 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 the 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 the remaining 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 the 2nd round of path I, the event mentioned in Proposition 2 happens again. We can use the saved table again to check if the candidates for the key lower nibbles can pass this round.

Because the values of input lower nibbles for one of the MixColumns in the first round are known for both plaintexts, we can guess their values after MixColumn (We will guess the value of four nibbles after the MixColumns transformation. 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 candidate subkey is equal to the saved values in table.

A plaintext pair and a candidate key can pass 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 in the path I. Alike previous step, time complexity of this step will be negligible compared to the time complexity of next step.

(5) **Inverting pairs of ciphertexts:** At this step we will invert every possible triplet of (C, C', k_{low}), generating possible pairs $(S, S')_r$ for $r = R, R-1, \ldots,$ (where $(S, S')_r$ represents the value of lower nibbles entering round r).

Since for every MixColumn there are 3 information bits, inverting one round costs 2^3 round encryptions per triplet. During the iterative rounds, the number of possible triplets 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 the 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 the 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 much 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 triplets multiplied by 2^3 (cost of inverting one round), multiplied by the number of iterative rounds. The cost for inverting non-iterative rounds is so small, because the number of candidates has been reduced so much. Time complexity for the other steps is 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 ciphertexts and their corresponding plaintexts can match the condition in previous step, we will find the whole bits of the master key with an exhaustive search for higher nibbles.

□

4.1. Results and complexities

Applying described key recovery attack to both paths, we are able to attack up to 14 and 15 rounds KLEIN which outperforms the cases introduced in [7]. Results of our attacks are shown in Table 2. The memory complexity is 2^{32} of block size in all of our 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	$2^{-94.49}$	$2^{92.08}$	$2^{63.49}$

As it can be seen in 2, using path I makes a good time complexity and path II makes a good data complexity. There is a trade-off between time and data. Regarding the attacks with the same number of rounds, our attacks have better time complexity or need less data than attacks in [7] (and in some cases both). But, they have greater memory complexity. Also, except biclique attacks, cryptanalyses of 14-round KLEIN-80 and 15-round KLEIN-96 were introduced for the first time in this paper.

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 was proposed by L a l l e m a n d and N a y a- P l a s e n c i a [7]. Results show that our attacks have the best time and data complexities on full-round KLEIN-64, 13-round KLEIN-80 and 14-round KLEIN-96 so far. Also, we introduced two new attacks on 14-round KLEIN-80 and 15-round KLEIN-96 for the 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 input nibbles, as the only transformation responsible to do that. (2) The Key Schedule does not mix higher and lower nibbles. These two enables the cryptanalyst to perform a reduced partial key search. Anyway, by using an appropriate diffusion layer instead of Rijndael's and a stronger Key Schedule one can prevent the attacks applied on KLEIN.

REFERENCES

[1] GONG, Z.—NIKOVA, S.—LAW, Y. W.: *KLEIN: A new family of lightweight block ciphers*, in: 7th Internat. Workshop on RFID Security and Privacy—RFIDSec '12 (A. Juels and Ch. Paar, eds.), Amherst, MA, USA, 2011, Lecture Notes in Math., Vol. 7055, Springer--Verlag, Berlin, 2012, pp. 1–18.

[2] YU, X.—WU, W.—LI, Y.—ZHANG, L.: *Cryptanalysis of reduced-round KLEIN block cipher*, in: 7th Internat. Conf. on Information Security and Cryptology—Inscrypt '12 (Ch.-K. Wu et al., eds.), Beijing, China, 2011, Lecture Notes in Math., Vol. 7537, Springer--Verlag, Berlin, 2012, pp. 237–250.

[3] AUMASSON, J. P.—NAYA-PLASENCIA, M.—SAARINEN, M. J. O.: *Practical attack on 8 rounds of the lightweight block cipher KLEIN*, in: 12th Internat. Conf. on Progress in Cryptology—INDOCRYPT '11 (D. J. Bernstein and S. Chatterjee, eds.), Chennai, India, 2011, Lecture Notes in Math., Vol. 7107, SpringerSpringer-Verlag, Berlin, 2011, pp. 134–145.

[4] AHMADIAN, Z.—SALMASIZADEH, M.—AREF, M. R.: *Biclique cryptanalysis of the full-round KLEIN block cipher*, IET Inform. Sec. J. **9** (2015), 294–301.

[5] ABED, F.—FORLER, C.—LIST, E.—LUCKS, S.—WENZEL, J.: *Biclique cryptanalysis of PRESENT, LED, and KLEIN*, Cryptology ePrint Archive, Report 2012/591, 2012.

[6] NIKOLIĆ, I.—WANG, L.—WU, SH.: *The parallel-cut meet-in-the-middle attack*, Cryptology ePrint Archive, Report 2013/530, 2013.

[7] LALLEMAND, V.—NAYA-PLASENCIA, M.: *Cryptanalysis of KLEIN*, in: 21st Internat. Workshop on Fast Software Encryption—FSE '14 (C. Cid and Ch. Rechberger, eds.), London, UK, 2014, Lecture Notes in Math., Vol. 8540, Springer-Verlag, Berlin, 2015, pp. 451–470.

[8] KNUDSEN, L. R.: *Truncated and higher order differentials*, in: 2nd Internat. Workshop on Fast Software Encryption—FSE '94 (B. Preneel, ed.), Leuven, Belgium, Lecture Notes in Math., Vol. 1008, Springer-Verlag, Berlin, 1994 pp. 196–211.

Received August 15, 2016

Shahram Rasoolzadeh
Simula Research Laboratory
Bergen
NORWAY
Lehrstuhl Embedded Security
Gebäude ID 2/607
Ruhr-Universität Bochum
Universitätsstraße 150
44801 Bochum
GERMANY
E-mail: shahram.Rasoolzadeh@uib.no
shahram.Rasoolzadeh@ruhr-uni-bochum.de

Zahra Ahmadian
Department of Electrical Engineering
Shahid Beheshti University
Daneshju Bulevard, District 1
Tehran
IRAN
E-mail: z_ahmadian@sbu.ac.ir

Mahmoud Salmasizadeh
Mohammad Reza Aref
Sharif Technical University
Azadi Ave
Tehran
IRAN
E-mail: salmasi@sharif.edu
aref@sharif.edu