Differential Fault Analysis on Midori

Wei Cheng[1,2], Yongbin Zhou[1,2(✉)], and Laurent Sauvage[3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, No. 89A Minzhuang Road, Beijing 100093, China
{chengwei,zhouyongbin}@iie.ac.cn
[2] University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing 100049, China
[3] Institute Mines-Tlcom, Telecom ParisTech, CNRS LTCI, Paris, France
laurent.sauvage@telecom-paristech.fr

Abstract. Midori is an energy-efficient lightweight block cipher published by Banik et al. in ASIACRYPT 2015, which consists of two variants with block sizes of 64-bit and 128-bit, respectively. In this paper, a new method is proposed to exploit cell-oriented fault propagation patterns in recognizing appropriate faulty ciphertexts and fault positions, which poses a serious threat to practical security of Midori. In light of this, we present a Differential Fault Attack against the Midori using cell-oriented fault model. Specifically, by inducing two random cell faults into the input of the antepenultimate round, our attack reduces the secret key search space from 2^{128} to 2^{32} for Midori-128 and from 2^{128} to 2^{80} for Midori-64, respectively. Our experiments confirmed that two faulty ciphertexts induced into the input of antepenultimate round could recover twelve in sixteen cells of subkey with over 80% probability.

Keywords: Lightweight cipher · Differential fault analysis · Cell-oriented fault propagation · Midori

1 Introduction

In recent years, Internet of Things (IoT) as a new information network technology is booming, accompanied by an endless stream of new devices including Smart-Home devices, wearable devices, medical implants and other battery operated portable equipments. These devices always produce, process, transfer and store private information such as wearable equipments, or even security-critically control over people's lives like heart pacemakers. Inevitably, there is growing concern about their actual security. Fortunately, the cryptographic technique is a reliable way to meet these security requirements. As a result, the resource-constrained devices like RFID tags and sensing nodes used in the IoT have drawn a great attention on the lightweight cipher, which is featured with low latency, small areas, low energy consumption and hardware-friendly design. In this flourishing field, several lightweight block ciphers have been proposed in the last few years, including HIGHT [1], CLEFIA [2], PRESENT [3], KATAN [4], PRINCE [5], LED [6], Piccolo [7], SIMON/SPECK [8], Midori [9] and so on.

K.-Y. Lam et al. (Eds.): ICICS 2016, LNCS 9977, pp. 307–317, 2016.
DOI: 10.1007/978-3-319-50011-9_24

Midori is published in the ASIACRYPT 2015 by Banik [9] et al. with two variants Midori-64 and Midori-128, both of them optimized with the energy consumption criterion. The optimizing work mainly consist of replacing the 8-bit Sboxes with 4-bit Sboxes and using almost MDS (Maximum Distance Separable) binary matrices instead of MDS matrices. By adopting this energy-efficient architecture, Midori seems to be a promising cipher with low latency and small areas at the same time. However, the security (mathematical security and practical security) of lightweight cipher is vital as it is the key to protect our security-sensitive information inside the IoT devices from attackers. On one hand, several literatures have studied the mathematical security of Midori by means of classical cryptanalysis, including differential/linear cryptanalysis [9], impossible differential cryptanalysis [10], meet-in-middle attack [11], truncated differential and related-key differential attacks [12]. Nonetheless, these cryptanalysis haven't identified any serious weakness with respect to mathematical properties. On the other hand, the practical security also plays a key role for security, but for Midori, there is no public literature studied its practical security so far.

Other than classical cryptanalysis, differential fault attack (DFA) is a typical cryptanalysis on cryptographic devices (implementations). It was first proposed by Biham and Shamir [13] against DES-like cryptosystem. After that, several similar attacks have been proposed to analyze the AES [14–16], Triple-DES [17], SMS4 [18,19], LED [20] et al. In essence, the DFA exploits the subtle relationships between the secret key and the behavior information under malfunctions to launch a key recovery attack. Typically, it derives information about the secret key by the differential between correct and faulty ciphertext (with the same plaintext). Thus, besides selecting a suitable fault model, the key to a DFA in practical is to determine whether the success of fault injection. All these aforementioned DFA methods haven't pointed out how to filter the proper faulty ciphertexts. As the DFA method described in [16], although it only need one faulty ciphertext to recover the 128-bit secret key, there is a huge difficulty to discriminate and sort out the applicable faulty ciphertext. On the other hand, the determination of the fault location also influences the analysis complexity. If it is unknown, the exhaustive method is needed to cover all possibilities, thus the computational complexity will be multiplied. Therefore, if the precise positions could be deduced straightly by the faulty and correct ciphertexts, the attacking complexity would be decreased dramatically. For Midori, the almost MDS matrix used in its permutation-layer gives us an unexpected convenience to solve both filtering and positioning problems, thus from a security point of view, this feature poses the great threat to its practical security against attacks like DFAs.

In this paper, we firstly illustrate an crucial vulnerability in Midori caused by the almost MDS binary matrix. We begin with investigating the fault propagation property of single fault induced in the antepenultimate round of encryption, then examine the differential of correct and faulty ciphertext, and analyze the positions of nonzero differentials. Some distinct patterns emerge, which connects the faulty position and the nonzero-differential positions, and these patterns could be exploited to deduce the corrupted positions. This fact also suggests

that the tradeoffs must be taken between security and the performance metrics like latency, energy consumption by (lightweight) cipher designers. Based on this observation, we propose a new cell-oriented differential fault analysis method against both Midori-64 and Midori-128, as they adopt the same overall structure. Our attack adopts cell-oriented fault model, and the fault injection position could be inferred only using correct and faulty ciphertext. By retrieving the related subkeys, our method reduces the secret key search space by 2^{48} and 2^{96} only using two faulty ciphertexts for Midori-64 and Midori-128, respectively.

The rest of this paper is organized as follows. Section 2 briefly introduces the block cipher Midori. Section 3 investigates the cell-oriented fault propagation of Midori. Then Sect. 4 proposes our DFA method, and Sect. 5 summarizes the attacking complexity and experiments. Finally Sect. 6 concludes the paper.

2 Description of Block Cipher Midori

Midori consists of two variants, Midori-64 and Midori-128. Their block sizes n are 64-bit and 128-bit respectively, and the key sizes are 128-bit for both. Midori adopts a typical Substitution-Permutation Network structure, its state matrix is a 4×4 cell-matrix, where the cell sizes m are 4-bit and 8-bit for Midori-64 and Midori-128, respectively. The state matrix S is defined as follows:

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix},$$

where $s_0, s_1, \ldots, s_{15}$ are sixteen cells. Midori is composed of encryption, decryption and key schedule, its overall structure of encryption is depicted as Fig. 1. And the comparison of two variants of Midori is tabulated as Table 1.

Table 1. The comparison of two Midori variants

	Block size(n)	Cell size(m)	Key size	Number of rounds(R)
Midori-64	64	4	128	16
Midori-128	128	8	128	20

2.1 Encryption and Decryption

For Midori, its encryption and decryption consist of R rounds of round function. Each of it consists of four transformations including $SubCell$, $ShuffleCell$, $MixColumn$ and $KeyAdd$. The plaintext is divided into 16 cells and rearranged into the state matrix S. The overall structure of encryption is pictured as Fig. 1 and these four transformations are described in the following.

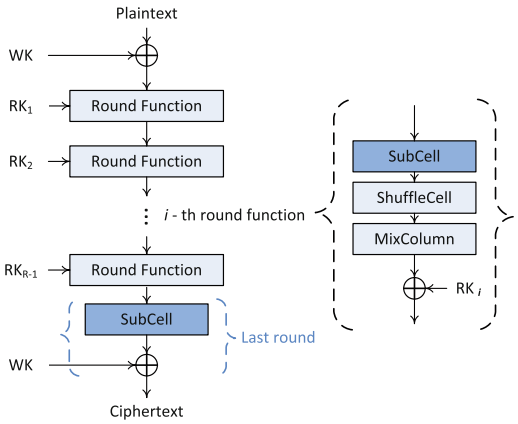

Fig. 1. Overall structure of Midori (Encryption), $R = 16$ for Midori-64 and $R = 20$ for Midori-128. WK is the whitening key and RK_i is the round key.

- $SubCell(SuC)$: For Midori-64, apply the 4-bit Sb_0 to the state matrix S for each cell: $s_i \leftarrow Sb_0(s_i)$. Similarly, for Midori-128, the 4-bit Sb_0 is replaced by 8-bit Sboxes (composed by two 4-bit Sb_1 and two bit-permutations [9]): SSb_0, SSb_1, SSb_2, SSb_3, namely, $s_i \leftarrow SSb_{i \bmod 4}(s_i)$, where $0 \leq i \leq 15$.
- $ShuffleCell(ShC)$: Each cell of the state S is permuted as follows: $(s_0, s_1, \ldots, s_{15}) \leftarrow (s_0, s_{10}, s_5, s_{15}, s_{14}, s_4, s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8)$
- $MixColumn(MC)$: M is applied to every $4m$-bit column of the state matrix S, i.e., ${}^t(s_i, s_{i+1}, s_{i+2}, s_{i+3}) \leftarrow M \cdot {}^t(s_i, s_{i+1}, s_{i+2}, s_{i+3})$ and $i = 0, 4, 8, 12$. Here M and its inverse matrix M' are defined as:

$$M = M' = \begin{pmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}$$

- $KeyAdd(AK)$: The i-th n-bit round key RK_i is XORed to a state matrix S.

Table 2. 4-bit bijective Sbox Sb_0 and Sb_1 in hexadecimal form [9]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sb_0	C	A	D	3	E	B	F	7	8	9	1	5	0	2	4	6
Sb_1	1	0	5	3	E	2	F	7	D	A	9	B	C	8	4	6

The decryption procedure shares the whole structure with encryption except that the $ShuffleCell$ is replaced by its inverse and the order of the round keys RK_i is from R to 0 (Fig. 1).

2.2 Key Schedule

For Midori-64, the 128-bit secret key K is the concatenation of two 64-bit keys K_0 and K_1, thus the $WK = K_0 \oplus K_1$ and the round key $RK_i = K_{(i-1) \ mod \ 2} \oplus \alpha_{i-1}$, where $1 \leq i \leq 15$. For Midori-128, $WK = K$ and $RK_i = K \oplus \beta_{i-1}$, where $1 \leq i \leq 19$. Note that α_i, β_i are both constants, and $\alpha_i = \beta_i$ for $0 \leq i \leq 14$.

2.3 Notations

The following notations were used throughout the rest of the paper.

- X^i, X^i_j: X^i is the output of the (i)-th round, $i = 1, 2, \ldots, R$, thus X^0 is defined as the plaintext and $X^R = C$ is the ciphertext. X^i_j is the j-th cell of X^i, $j = 0, 1, \ldots, 15$.
- SuC^i, ShC^i, MC^i, AK^i: these are the state matrix after $SubCell$, $ShuffleCell$, $MixColumn$ and $KeyAdd$ of the i-th round, respectively. Namely, $AK^R = X^R$ is the ciphertext.
- RK^i: the round key used in the i-th round function, $i = 1, 2, \ldots, R-1$, and $RK^0 = WK,\ RK^R = WK$.
- ΔX: the difference of two state matrixes X and X'.

3 Cell-Oriented Fault Propagation Analysis

In this section, we investigate the propagation of one cell-oriented fault induced into the input of (R-2)-th round function.

3.1 Fault Propagation in Last Three Rounds

Due to the simple diffusion pattern of inducing fault into the input of last round and penultimate round, we focus on the single cell-oriented random fault injected into the input of antepenultimate round. As depicted in Fig. 2, the single cell fault f induced before the antepenultimate round is changed into f' after $SubCell$, and remains unchanged after $ShuffleCell$, then the other three cells in the same column are infected with identical differential f', which stay the same after $KeyAdd$ transformation. The refreshing of differential values in the (R-1)-th round is similar to (R-2)-th round, and then trivial in R-th round because of omitted permutation-layer in the last round. Thus the output differentials equal to the XOR of correct ciphertext and faulty ciphertext or two faulty ciphertexts.

3.2 Cell-Oriented Fault Propagation Patterns

Distinct association patterns could be observed between the positions of nonzero-differentials and the position of single cell-oriented fault. Specifically, as pictured in Fig. 3, each cell of faults induced in the input of antepenultimate round results in nine faulty cells with unique patterns. Apparently, these patterns could be

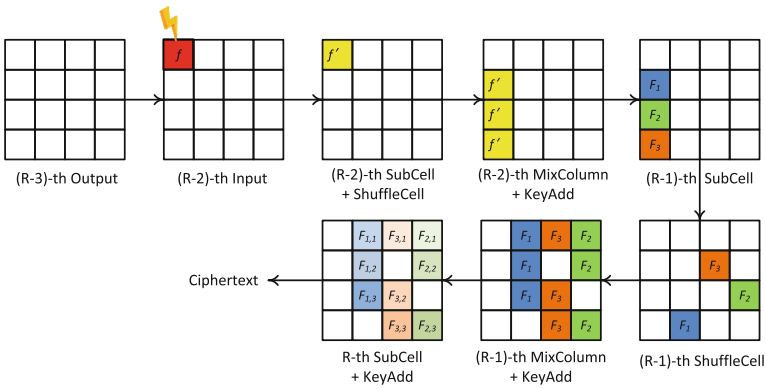

Fig. 2. The fault propagation of last three rounds, fault is induced into the first cell of state matrix. f, f', F_i, and $F_{i,j}$ are the differentials of two corresponding intermediates, where $i, j = 1, 2, 3$.

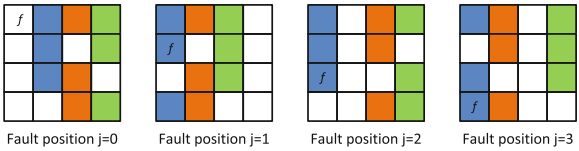

Fig. 3. The fault propagation patterns of four cells in the first column, f is the faulty differential and its position is corresponding to the fault injection position.

straightforwardly inferred only requiring the correct ciphertext and faulty ciphertext. That is, the position of corrupted cell in state matrix S could be uniquely determined. The essential reason of this pattern is caused by the almost MDS matrix applied in the permutation layer of Midori. The fault propagation patterns of four cells in the first state matrix column are depicted as Fig. 3 (fault position $\mathrm{j} = 0$ is detailed in Fig. 2).

In Midori, all design choices are made to save energy consumption, including using 4×4 almost MDS binary matrices instead of 4×4 MDS matrices. However, because the branch number of almost MDS matrices [9] is 4, one nonzero active input leads to three nonzero active outputs in the same column, and nine nonzero active outputs after two rounds transformations. As a result, the diffusing effect is weak enough to find distinct patterns of faulty positions. Compared to MDS matrix applied in the AES (Advanced Encryption Standard), after two rounds of round function, one nonzero active input leads to sixteen nonzero active outputs, therefore, no obvious association patterns between the fault injection position and nonzero differential positions in output.

In the view of practical security, this energy-efficient almost MDS matrix gives rise to a vulnerability, especially faced with differential fault attacks. This fact demonstrates that the tradeoffs must be taken between security and the performance metrics like latency, energy consumption by (lightweight) cipher

designers. For practical security, the full diffusion MDS matrix is evidently more preferable than almost MDS matrix for SPN-structure ciphers.

4 Cell-Oriented Differential Fault Analysis on Midori

As pictured in Fig. 2, the fault propagation patterns are clear. Let us denote the correct ciphertext $C = X^R = AK^R$ and faulty ciphertext $C' = X'^R = AK'^R$, then we have the differential $\triangle C$ that

$$\triangle C = C \oplus C' = X^R \oplus X'^R \tag{1}$$

For the (R-1)-th round, the output differential has three nonzero values which equal to F_1, F_2, F_3, respectively. Thus, due to the involution property of Sboxes applied in the Midori, for F_1, we have following four equations:

$$\begin{aligned} F_1 &= SuC(C_4 \oplus RK_4^R) \oplus SuC(C'_4 \oplus RK_4^R) \\ F_1 &= SuC(C_5 \oplus RK_5^R) \oplus SuC(C'_5 \oplus RK_5^R) \\ F_1 &= SuC(C_6 \oplus RK_6^R) \oplus SuC(C'_6 \oplus RK_6^R) \\ 0 &= SuC(C_7 \oplus RK_7^R) \oplus SuC(C'_7 \oplus RK_7^R) \end{aligned} \tag{2}$$

where $F_1 \in \mathbb{F}_{2^m}$, $m = 4$ for Midori-64 and $m = 8$ for Midori-128. These four equations can be solved for three subkey cells RK_4^R, RK_5^R, RK_6^R. The key search space of this triple of key cells is reduced to an expected value of 2^m from $(2^m)^3 = 2^{3m}$. Similar equations could be deduced for F_2 and F_3, thus after combination of these three classes of equations, the whole key search space related to F_1, F_2 and F_3 is reduced to 2^{3m} from 2^{9m}.

By continuing this method, and X^{R-2} is the output of (R-2)-th round, then its first column is:

$$\begin{aligned} X_0^{R-2} &= SuC(SuC(C_1 \oplus RK_1^R) \oplus RK_1^{R-1} \oplus SuC(C_2 \oplus RK_2^R) \\ &\quad \oplus RK_2^{R-1} \oplus SuC(C_3 \oplus RK_3^R) \oplus RK_3^{R-1}) \\ X_1^{R-2} &= SuC(SuC(C_4 \oplus RK_4^R) \oplus RK_4^{R-1} \oplus SuC(C_5 \oplus RK_5^R) \\ &\quad \oplus RK_5^{R-1} \oplus SuC(C_6 \oplus RK_6^R) \oplus RK_6^{R-1}) \\ X_2^{R-2} &= SuC(SuC(C_{12} \oplus RK_{12}^R) \oplus RK_{12}^{R-1} \oplus SuC(C_{13} \oplus RK_{13}^R) \\ &\quad \oplus RK_{13}^{R-1} \oplus SuC(C_{15} \oplus RK_{15}^R) \oplus RK_{15}^{R-1}) \\ X_3^{R-2} &= SuC(SuC(C_8 \oplus RK_8^R) \oplus RK_8^{R-1} \oplus SuC(C_{10} \oplus RK_{10}^R) \\ &\quad \oplus RK_{10}^{R-1} \oplus SuC(C_{11} \oplus RK_{11}^R) \oplus RK_{11}^{R-1}) \end{aligned} \tag{3}$$

Thus considering its cell-oriented differentials in the output of (R-2)-th round:

$$\begin{aligned} 0 &= \triangle X_0^{R-2} = X_0^{R-2} \oplus X_0'^{R-2} \\ f' &= \triangle X_1^{R-2} = X_1^{R-2} \oplus X_1'^{R-2} \\ f' &= \triangle X_2^{R-2} = X_2^{R-2} \oplus X_2'^{R-2} \\ f' &= \triangle X_3^{R-2} = X_3^{R-2} \oplus X_3'^{R-2} \end{aligned} \tag{4}$$

Consequently, the interrelation between subkey cells in the equations further reduce the subkey search space. Apparently, for the first cell output of (R-3)-th round and its nonzero differential are:

$$\begin{aligned} X_0^{R-3} &= SuC(X_1^{R-2} \oplus RK_1^{R-2} \oplus X_2^{R-2} \oplus RK_2^{R-2} \oplus X_3^{R-2} \oplus RK_3^{R-2}) \\ f &= X_0^{R-3} \oplus X_0'^{R-3} \end{aligned} \tag{5}$$

With combination of equations Eqs. 2, 4 and 5, only twelve cells of subkey essentially involve nonzero differential operations, resulting that the key search space is reduced to an expected decrease value of 2^{12m}.

5 Attacking Complexity and Experimental Results

5.1 Attacking Complexity

In essence, differential fault analysis utilizes the interrelationship of input differential and output differential in the *SubCell*. For Midori, its relationship is defined as follows [18]:

$$\begin{aligned} INs(\triangle x, \triangle y) &= \{zi | zi \in \mathbb{F}_{2^m}, SuC(zi) \oplus SuC(zi \oplus \triangle x) = \triangle y\} \\ Ns(\triangle x, \triangle y) &= \#\{zi | zi \in \mathbb{F}_{2^m}, SuC(zi) \oplus SuC(zi \oplus \triangle x) = \triangle y\} \end{aligned} \tag{6}$$

then for Midori's last round *SubCell*, using the first equation of Eq. 2, $\triangle x = C_4 \oplus C_4'$, $\triangle y = F_1$, then the candidates of subkey cell could be recovered using $RK_4^R = C_4 \oplus INs$. Candidates of other subkey cells could be derived similarly.

Note that the maximum differential probability [9] of *SubCell* are 2^{-2}, namely, maximum of Ns equals to $16 \times 2^{-2} = 4$ for Midori-64 and $256 \times 2^{-2} = 64$ for Midori-128. That is, for fixed $\triangle x, \triangle y$, the maximum number of subkey cell candidates should be 4 and 64 for Midori-64 and Midori-128, respectively. Specifically, for Sb_0, Sb_1 separately used in Midori-64 and Midori-128, if $Ns(\triangle x, \triangle y)$ is not null, then it equals 2 with probability of 75.0% (72/96) and 85.71% (90/105) for Sb_1. Due to the *SubCell* of Midori-128 is constructed by Sb_1, if it is divided into two of Sb_1, the attacking complexity could be reduced dramatically.

Since at least two faults are required to uniquely determine the subkey cell candidates in equations Eq. 2, we derive intersection of subkey cell candidates using multiple faults induced in the same rounds (optional). Given that two faults are induced in the same cell position of (R-2)-th input, three nonzero differentials are obtained by pairing combination. Therefore, for Midori-128, at least two faulty ciphertexts are required to recover nine cells of RK^R and three cells of RK^{R-1}. Considering that $RK^{R-1} = RK^{R-1} \oplus \beta_{R-2}$ and $K = WK = RK^R$, hence twelve cells of secret key K could be deduced, its secret key search space is reduced to 2^{32} from 2^{128} at best. For Midori-64, two faulty ciphertexts could only recover nine cells of RK^R and three cells of RK^{R-1}, thus secret key search space decreases by an expected value of 2^{48} $(= 2^{12m})$.

5.2 Experimental Results

We implemented our attack on a PC using Matlab R2014b (64-bit) with 2.60 GHz CPU and 4 GB memory. The fault injection was simulated by software commands. We use the equations similar to Eq. 2 to illustrate our attack which is applied to Midori-64. Two simulated faults were induced into the first cell of (R-1)-th input, namely X_0^{R-2} and the corrupted value is kept unknown.

Table 3. Subkey cell recovery for RK_1^R, RK_2^R and RK_3^R using two faulty ciphertexts

	cNum = 1	cNum = 2	cNum = 4	Proportion of cNum = 1	Time latency(s)
RK_1^R	92	4	9	87.62%	0.2381
RK_2^R	86	10	9	81.90%	0.2389
RK_3^R	86	10	9	81.90%	0.2629

*cNum = #{Candidates} denotes the number of subkey cell candidates. The first three columns denote the number of possible combinations of two distinct differentials satisfying cNum = 1, cNum = 2 and cNum = 3, respectively.

Considering that $Ns(\triangle x, \triangle y)$ of Sb_0, the number of nonzero input $\triangle x$ equals 15, thus all combinations of two distinct differentials only have 105 ($= 15 \times 14/2$) elements. In consequence, as tabulated in Table 3, three subkey cells of RK^R could be recovered with over 80% probability only using two faulty ciphertexts.

On the basis of above experiments, for Midori-128, the attacking complexity in practice is estimated to $2^9 \cdot (3e^2 + 3e)$ ($= 12 \cdot C_{(e+1)}^2 \cdot 2^8$), where e denotes the number of faults induced in the same cell position. For Midori-64, with the same setting, the attacking is estimated to $2^5 \cdot (3e^2 + 3e)$.

6 Conclusions

In this paper, based on the cell-oriented fault propagation patterns existing in Midori, we presented a differential fault analysis method against its two variants Midori-64 and Midori-128. Our method straightly exploits these patterns to uniquely determine the corrupted positions, resulting in its low attacking complexity. Especially, secret key search space is reduced from 2^{128} to 2^{32} for Midori-128 and from 2^{128} to 2^{80} for Midori-64, respectively. In addition, our experimental results confirms that the almost MDS matrix used in its permutation layer resulting in a vulnerability, which could be utilized by practical attacks like DFAs. This result evidently provides a new design advice to cipher designers.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China (Grant No. 61272478, No. 61472416 and No. 61632020) and Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA06010701 and XDA06010703).

References

1. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: 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). doi:10.1007/11894063_4
2. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit blockcipher CLEFIA (extended abstract). In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Heidelberg (2007). doi:10.1007/978-3-540-74619-5_12
3. Bogdanov, 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). doi:10.1007/978-3-540-74735-2_31
4. Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04138-9_20
5. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34961-4_14
6. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23951-9_22
7. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: *Piccolo*: an ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23951-9_23
8. Ray, B., Douglas, S., Jason, S., Stefan, T.-C., Bryan, W., Louis, W.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404 (2013). http://eprint.iacr.org/
9. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: a block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 411–436. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48800-3_17
10. Cheng, Z., Wang, X.: Impossible Differential Cryptanalysis of Midori. Cryptology ePrint Archive, Report 2016/535 (2016). http://eprint.iacr.org/
11. Lin, L., Wu, W.: Meet-in-the-Middle Attacks on Reduced-Round Midori-64. Cryptology ePrint Archive, Report 2015/1165 (2015). http://eprint.iacr.org/
12. Dong, X., Shen, Y.: Cryptanalysis of Reduced-Round Midori64 Block Cipher. Cryptology ePrint Archive, Report 2016/676 (2016). http://eprint.iacr.org/
13. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991). doi:10.1007/3-540-38424-3_1
14. Giraud, C.: DFA on AES. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) AES 2004. LNCS, vol. 3373, pp. 27–41. Springer, Heidelberg (2005). doi:10.1007/11506447_4
15. Mukhopadhyay, D.: An improved fault based attack of the advanced encryption standard. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 421–434. Springer, Heidelberg (2009). doi:10.1007/978-3-642-02384-2_26

16. Tunstall, M., Mukhopadhyay, D., Ali, S.: Differential fault analysis of the advanced encryption standard using a single fault. In: Ardagna, C.A., Zhou, J. (eds.) WISTP 2011. LNCS, vol. 6633, pp. 224–233. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21040-2_15
17. Hemme, L.: A differential fault attack against early rounds of (triple-)DES. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 254–267. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28632-5_19
18. Li, R., Sun, B., Li, C., You, J.: Differential fault analysis on SMS4 using a single fault. J. Inf. Process. Lett. **111**(4), 156–163 (2011)
19. Wu, W., Zhang, L.: Differential fault analysis on SMS4. J. Int. J. Comput. Intell. Syst. **9**, 011 (2006)
20. Li, W., Gu, D., Xia, X., Zhao, C., Liu, Z., Liu, Y., Wang, Q.: Single byte differential fault analysis on the LED lightweight cipher in the wireless sensor network. J. Int. J. Comput. Intell. Syst. **5**(5), 896–904 (2012)