SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2018, Vol. 61 032110:1–032110:13
https://doi.org/10.1007/s11432-017-9209-0

Impossible meet-in-the-middle fault analysis on the LED lightweight cipher in VANETs

Wei LI[1,2,3,4], Vincent RIJMEN[2], Zhi TAO[1], Qingju WANG[4,5,2], Hua CHEN[6,2], Yunwen LIU[2,7], Chaoyun LI[2] & Ya LIU[8,4]*

[1]*School of Computer Science and Technology, Donghua University, Shanghai* 201620*, China;*
[2]*imec-COSIC, KU Leuven, Leuven* 3000*, Belgium;*
[3]*Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai* 200240*, China;*
[4]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[5]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs. Lyngby* 2800*, Denmark;*
[6]*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[7]*College of Science, National University of Defense Technology, Changsha* 410073*, China;*
[8]*Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai* 200093*, China*

Received 10 May 2017/Accepted 19 July 2017/Published online 12 January 2018

Abstract With the expansion of wireless technology, vehicular ad-hoc networks (VANETs) are emerging as a promising approach for realizing smart cities and addressing many serious traffic problems, such as road safety, convenience, and efficiency. To avoid any possible rancorous attacks, employing lightweight ciphers is most effective for implementing encryption/decryption, message authentication, and digital signatures for the security of the VANETs. Light encryption device (LED) is a lightweight block cipher with two basic keysize variants: LED-64 and LED-128. Since its inception, many fault analysis techniques have focused on provoking faults in the last four rounds to derive the 64-bit and 128-bit secret keys. It is vital to investigate whether injecting faults into a prior round enables breakage of the LED. This study presents a novel impossible meet-in-the-middle fault analysis on a prior round. A detailed analysis of the expected number of faults is used to uniquely determine the secret key. It is based on the propagation of truncated differentials and is surprisingly reminiscent of the computation of the complexity of a rectangle attack. It shows that the impossible meet-in-the-middle fault analysis could successfully break the LED by fault injections.

Keywords VANETs, LED, lightweight cipher, impossible meet-in-the-middle, fault analysis

Citation Li W, Rijmen V, Tao Z, et al. Impossible meet-in-the-middle fault analysis on the LED lightweight cipher in VANETs. Sci China Inf Sci, 2018, 61(3): 032110, https://doi.org/10.1007/s11432-017-9209-0

1 Introduction

Vehicular ad-hoc networks (VANETs) are appearing as a new landscape of mobile ad-hoc networks, with the aim of providing a wide spectrum of safety and comfort applications for drivers and passengers. They have been tremendously successful, and have attracted considerable attention from both academia and industry [1]. However, VANETs are networks with high dynamic topology and their connections are vulnerable to attacks. For instance, attackers may exploit VANETs to send bogus information to

* Corresponding author (email: liuyaloccs@gmail.com)

deceive other vehicles. Therefore, security conservation in VANETs is an indispensable demand. Nodes in VANETs should be confident that each instance of communication has been started from a trustworthy source node and messages are not modified by malicious vehicles. Although these issues seem similar to those in traditional communication networks, there are characteristics specific to VANETs. The seriousness of security failures, self-organized nature of networks, high mobility of vehicles, relevance of vehicles to their geographic position, and irregular connectivity between vehicles can cause different security issues in VANETs [2–4]. On the limitation of processing capability, power supply, and memory space of highly constrained devices in vehicles, traditional ciphers cannot play direct roles in many security applications, such as encryption/decryption, message authentication, and digital signatures. It is very serious and urgent to implement effective ciphers in VANETs, i.e., lightweight ciphers are primarily selected for confidentiality, authentication, and integrity [5–13]. Thus, the application of lightweight ciphers can reduce device energy consumption, and allow increased network communications with lower-resource devices in vehicles.

The lightweight encryption device (LED) can be optimized for the radio frequency identification (RFID) tags and other highly constrained vehicle security devices in VANETs [14]. Its security has been demonstrated by the designers against linear attack, differential attack, algebraic attack, cube tester, integral attack, rotational attack, and slide attack. Mendel et al. [15] improved a differential attack depending on the mega-boxes and super-boxes. Isobe et al. [16] applied a low key-dependency to the key schedule and presented a meet-in-the-middle attack on the internal rounds of the LED. Later, Nikolić et al. [17] made use of the multicollision and slidex attacks on a round-reduced version of the LED. Soleimany presented a probabilistic slide attack on LED-64 [18]. Except for the traditional cryptanalysis, much of the research in recent years focuses on the LED against fault analysis [19–25].

In the last two decades, a serious threat against cryptographic implementation was put forward by fault analysis. Fault analysis can deduce a secret key by applying the mathematical relations of a cipher resulting from correct and faulty operations. Boneh et al. [25,26] presented the RSA against fault analysis by provoking the faulty bits in 1996. Later, a multitude of fault analysis techniques, including differential fault analysis (DFA), impossible differential fault analysis (IDFA), and meet-in-the-middle fault analysis (MFA) were proposed for breaking block ciphers [27–32]. The attackers could inject faults into the running procedure by exploring a glitch in the clock, spike in the power supply, or by implementing the external processes of a laser and electromagnetic radiations. They take advantage of the leaked faulty calculations with mathematical methods. Usually, fault analysis is much stronger than traditional cryptanalysis.

As for LED, recent research of fault analysis has been devoted to deriving calculations regarding the secret key by examining the differential, algebraic, statistical, or impossible differential relations to recover the subkeys. Table 1 illustrates the comparison of the latest fault analysis results for the LED. Three research groups proposed DFA to break the LED in the same year [19–21]. They can break the last subkey by injecting faults into the antepenultimate round of the LED. Jeong et al. [19] managed to derive a 64-bit secret key using one random nibble fault injection. Li et al. [20] extended a random nibble-oriented fault model to a random byte-oriented fault model, and broke LED-64 and LED-128 with three and six faults, respectively. Jovanovic et al. [21] applied certain proportional relationship techniques between different layers to reduce the number of faults to one and two, respectively. Then, Zhao et al. [22] proposed an algebraic fault analysis (AFA) by inducing the same faults into the antepenultimate round. They used an algebraic relationship to describe the intermediate value of the LED. Based on a statistical relationship, Ghalaty et al. [23] presented a differential fault intensity analysis (DFIA) by introducing 14 and 28 biased faults into the last round of LED-64 and LED-128, respectively. In 2016, Li et al. [24] presented an IDFA on the LED and extended fault locations to the third-last round with 48 and 96 faults, respectively. Previous fault analysis only targets the last four rounds of the LED.

Adding protection to full rounds of a lightweight cipher is ideal against fault attacks for highly constrained devices in VANETs. However, it can decrease performance, and is usually expensive in many implementations. Hence, practical countermeasures are suggested to protect only the first and last several rounds of a cipher in these devices. In real applications, random faults can occur in any round or register of the lightweight cipher. It is excellent that the highly constrained devices can be resistant

Table 1 Summary of fault analysis on LED

Type	First fault location	♯Faults on LED-64	♯Faults on LED-128	Ref.
DFA	r-2	1	–	[19]
		3	6	[20]
		1	2	[21]
AFA	r-2	1	2	[22]
DFIA	r	14	28	[23]
IDFA	r-3	48	96	[24]
IMFA	r-4	44.2	88.4	This paper

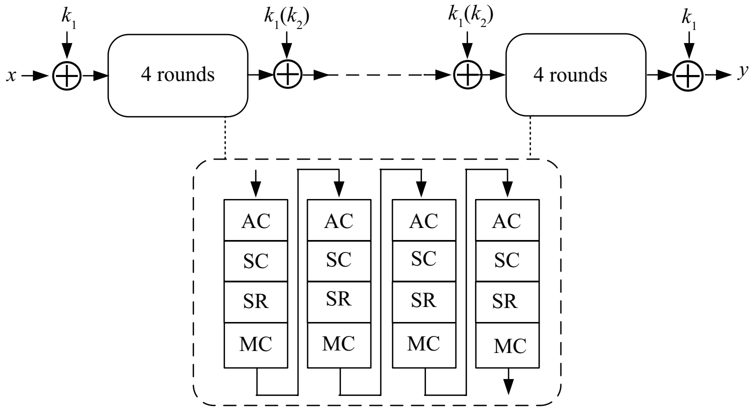

Figure 1 Structure of LED.

against all kinds of malicious attackers and dangerous environments. In other words, any vulnerability of a lightweight cipher against fault analysis should be detected as soon as possible, if fault locations can be extended to more rounds. This is our motivation for investigating novel fault analysis by attacking earlier rounds of the LED.

In this study, a novel impossible meet-in-the-middle fault analysis (IMFA) is successfully applied to break the LED. Compared with the previous fault analysis, faults can be injected into the fourth last round of the LED, and the novel fault path in IMFA affects more rounds. The attackers take advantage of the connection between an impossible relation and a meet-in-the-middle relation to recover the subkeys of the LED. Until now, the fault location was the deepest round of the LED cipher. Based on the propagation of truncated differentials, we present a detailed analysis to describe the attack complexity in a rectangle view. It can measure the connection of two different and independent relations, and thus, improve the theoretical accuracy.

The remainder of this paper is organized as follows: Section 2 describes the specification of the LED. Section 3 introduces the impossible differential fault analysis and meet-in-the-middle fault analysis, respectively. Then, Section 4 proposes our impossible meet-in-the-middle fault analysis for breaking LED-64 and LED-128. The next two sections present the attack complexity and analyze the experimental results. The last section concludes the paper.

2 Specification of the LED

The LED lightweight cipher fixes the block length to 64 bits, and supports key lengths of both 64 and 128 bits [14]. It has 32 and 48 rounds for LED-64 and LED-128, respectively, as Figure 1 shows. The state can be pictured as a rectangular array of nibbles, consisting of four rows and four columns. Each basic step is a sequence of four identical rounds with a subkey addition, denoted as AddRoundKey (ARK). Each round is composed of AddConstants, SubCells, ShiftRows, and MixColumnsSerial in sequence.

- AddConstants (AC) adds constants to the state with a bitwise XOR operation.
- SubCells (SC) applies S-boxes to each nibble of the state independently.

Table 2 Versions of LED

Version	Key size	Block size	Rounds	Key schedule
LED-64	64	64	32	$K = k_1$
LED-128	128	64	48	$K = k_1 \|\| k_2$

Table 3 Notations of LED

Notation	Description
x	The 64-bit plaintext
y, $\hat{y}$	The 64-bit correct and faulty ciphertexts
k_1, k_2	The 64-bit subkeys from the secret key K
r	The number of rounds with $r \in \{32, 48\}$
α_l, β_l, γ_l, δ_l	The 64-bit output of the AC, SC, SR, and MC layers in the l-th round with $1 \leqslant l \leqslant r$
$\hat{\alpha}_l$, $\hat{\beta}_l$, $\hat{\gamma}_l$, $\hat{\delta}_l$	The 64-bit faulty output of the AC, SC, SR, and MC layers in the l-th round with $1 \leqslant l \leqslant r$
β_r, $\hat{\beta}_r$	The values before addition with the correct subkey k_1, and $\beta_r = y \oplus k_1$, $\hat{\beta}_r = \hat{y} \oplus k_1$
g	The guess for k_1
z, $\hat{z}$	The values obtained by xoring the ciphertexts with the guess for the subkey, and $z = y \oplus g$, $\hat{z} = \hat{y} \oplus g$
μ, $\hat{\mu}$	The values derived from z in the same way as δ_{r-1} is derived from β_r
ω, $\hat{\omega}$	The values derived from μ in the same way as β_{r-1} is derived from δ_{r-1}
IAC, ISC, ISR, IMC	The inverse operation of the AC, SC, SR, and MC layers

- ShiftRows (SR) cyclically shifts each row of the state by different offsets.
- MixColumnsSerial (MC) takes all the columns and multiplies their data with a matrix.

The sequence of steps for the decryption is the same as that of the encryption using the same subkeys. The secret key, K, depends on a key schedule to generate two subkeys, k_1 and k_2, for the LED as shown in Table 2.

3 The IDFA and MFA attack on LED

3.1 Notations

The notations of LED and its analysis are described as shown in Table 3.

3.2 Fault model and main procedure

The fault model includes the chosen plaintext attacks and random nibble-oriented fault model. The IDFA and MFA are two independent types of fault analysis, which are proposed to attack AES [32]. Certain random faults are injected into the third last round of the running procedure, and thus, correct and faulty ciphertexts are obtained. Then, main procedures exploit the impossible relationship and meet-in-the-middle relation of the SubCells, respectively. As for the IDFA attack, the output differences in each nibble of the penultimate SubCells are not null. That is,

$$\begin{cases} (\beta_{r-1} \oplus \hat{\beta}_{r-1})^{4i} \neq 0, \\ (\beta_{r-1} \oplus \hat{\beta}_{r-1})^{4i+1} \neq 0, \\ (\beta_{r-1} \oplus \hat{\beta}_{r-1})^{4i+2} \neq 0, \\ (\beta_{r-1} \oplus \hat{\beta}_{r-1})^{4i+3} \neq 0, \end{cases}$$

where i represents the i-th column of the state, and $0 \leqslant i \leqslant 3$. As for the MFA attack, the input differences in each nibble of the penultimate SubCells have the following relations:

$$\begin{cases} (\alpha_{r-1} \oplus \hat{\alpha}_{r-1})^{4i} = \xi_{4i}, \\ (\alpha_{r-1} \oplus \hat{\alpha}_{r-1})^{4i+1} = \xi_{4i+1}, \\ (\alpha_{r-1} \oplus \hat{\alpha}_{r-1})^{4i+2} = \xi_{4i+2}, \\ (\alpha_{r-1} \oplus \hat{\alpha}_{r-1})^{4i+3} = \xi_{4i+3}, \end{cases}$$

where all vectors of $\{\xi_{4i}, \xi_{4i+1}, \xi_{4i+2}, \xi_{4i+3}\} \subseteq (\{0,1\}^4/\{0\})^4$ are proportional, and $0 \leqslant i \leqslant 3$. Thus, the last subkey can be recovered. Then, the attacker can recover the last subkey and decrypt the right ciphertext to obtain the input of the last round. They repeat the above procedure to induce faults to the running procedure until the secret key is derived. In [24], the IDFA attack recovered LED-64 and LED-128 with 48 and 96 faults, respectively. There are no experimental results regarding the MFA attack on the LED.

4 Impossible meet-in-the-middle fault analysis on LED

In the novel impossible meet-in-the-middle fault analysis, the attackers can store a ciphertext when encrypting any plaintext with a secret key. Their aim is to recover the subkey, k_1, in the last round. The first fault injection targets the (r–4)-th round, where $r \in \{32, 48\}$. As Figure 2 shows, a fault may be injected into α_{r-4}, β_{r-4} or γ_{r-4}; the approach is identical in either case. Any modification provokes the XOR-differences of the last five rounds, and the correct ciphertext, y, are converted into the faulty ciphertext, $\hat{y}$. The attackers have

$$\begin{aligned} \beta_r &= \mathrm{ISR}(\mathrm{IMC}(y \oplus k_1)) \\ &= \mathrm{ISR}(\mathrm{IMC}(y)) \oplus \mathrm{ISR}(\mathrm{IMC}(k_1)) \\ &= y' \oplus k_1', \end{aligned}$$

$$\begin{aligned} \hat{\beta}_r &= \mathrm{ISR}(\mathrm{IMC}(\hat{y} \oplus k_1)) \\ &= \mathrm{ISR}(\mathrm{IMC}(\hat{y})) \oplus \mathrm{ISR}(\mathrm{IMC}(k_1)) \\ &= \hat{y}' \oplus k_1', \end{aligned}$$

where

$$y' = \mathrm{ISR}(\mathrm{IMC}(y)), \quad \hat{y}' \ = \mathrm{ISR}(\mathrm{IMC}(\hat{y})), \quad k_1' = \mathrm{ISR}(\mathrm{IMC}(k_1)).$$

And

$$\begin{aligned} \delta_{r-2} \oplus \hat{\delta}_{r-2} =& \mathrm{IAC}(\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{IAC}(\mathrm{ISC}(\beta_r)))))) \oplus \mathrm{IAC}(\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{IAC}(\mathrm{ISC}(\hat{\beta}_r)))))) \\ =& \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{AC}(\mathrm{ISC}(y' \oplus k_1'))))) \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{AC}(\mathrm{ISC}(\hat{y}' \oplus k_1'))))). \end{aligned}$$

Because the output difference in each nibble of the antepenultimate SubCells and ShiftRows layers are not null, the impossible differential relationship must hold

$$(\gamma_{r-2} \oplus \hat{\gamma}_{r-2})^j = (\mathrm{IMC}(\delta_{r-2} \oplus \hat{\delta}_{r-2}))^j \neq 0,$$

where $0 \leqslant j \leqslant 15$. Thus, there are four groups of meet-in-the-middle relationships for every column of $\delta_{r-2} \oplus \hat{\delta}_{r-2}$ as follows:

$$\begin{cases} (\delta_{r-2} \oplus \hat{\delta}_{r-2})^{4i} = \varphi_\eta^{4i}, \\ (\delta_{r-2} \oplus \hat{\delta}_{r-2})^{4i+1} = \varphi_\eta^{(4i+13) \bmod 16}, \\ (\delta_{r-2} \oplus \hat{\delta}_{r-2})^{4i+2} = \varphi_\eta^{(4i+10) \bmod 16}, \\ (\delta_{r-2} \oplus \hat{\delta}_{r-2})^{4i+3} = \varphi_\eta^{(4i+7) \bmod 16}, \end{cases}$$

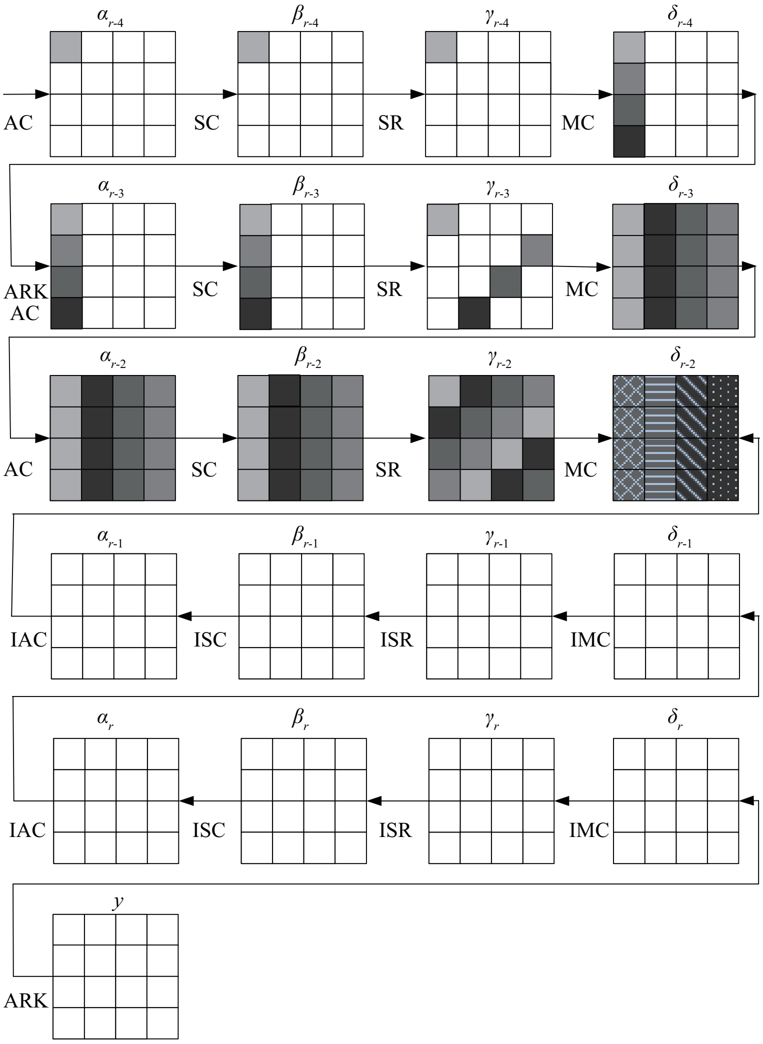

Figure 2 One of the fault attacking paths in the last five rounds.

where i represents the i-th column of the state, mod denotes the modular operation, φ_η represents all possible solutions of $(\gamma_{r-2} \oplus \hat{\gamma}_{r-2})^j \neq 0$, $0 \leqslant \eta \leqslant 15^4-1$, $0 \leqslant i \leqslant 3$ and $0 \leqslant j \leqslant 15$. Hence,

$$\begin{cases} \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_0(\mathrm{AC}(\mathrm{ISC}(y'^{4i} \oplus k_1'^{4i})))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_0(\mathrm{AC}(\mathrm{ISC}(\hat{y}'^{4i} \oplus k_1'^{4i}))))) = \varphi_\eta^{4i},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_1(\mathrm{AC}(\mathrm{ISC}(y'^{4i+1} \oplus k_1'^{4i+1})))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_1(\mathrm{AC}(\mathrm{ISC}(\hat{y}'^{4i+1} \oplus k_1'^{4i+1}))))) = \varphi_\eta^{(4i+13) \bmod 16},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_2(\mathrm{AC}(\mathrm{ISC}(y'^{4i+2} \oplus k_1'^{4i+2})))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_2(\mathrm{AC}(\mathrm{ISC}(\hat{y}'^{4i+2} \oplus k_1'^{4i+2}))))) = \varphi_\eta^{(4i+10) \bmod 16},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_3(\mathrm{AC}(\mathrm{ISC}(y'^{4i+3} \oplus k_1'^{4i+3})))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_3(\mathrm{AC}(\mathrm{ISC}(\hat{y}'^{4i+3} \oplus k_1'^{4i+3}))))) = \varphi_\eta^{(4i+7) \bmod 16}, \end{cases}$$

where $0 \leqslant i \leqslant 3$. The above equations allow the restriction of possible candidates for k_1'. The attackers can perform a brute-force search for k_1', column per column, until there is only one left in the set of k_1' candidates by intersections. Thus, the equation can be solved for K in LED-64:

$$K = k_1 = \mathrm{MC}(\mathrm{SR}(k_1')).$$

As for LED-128, the attackers can decrypt the last four rounds using the subkey k_1, to obtain the input of the (r–3)-th round, represented as α_{r-3}. They can take the above attack procedure to derive all nibbles of k'_2 when random faults are injected before δ_{r-8} in the (r–8)-th round. They have

$$\begin{aligned}\beta_{r-4} &= \mathrm{ISR}(\mathrm{IMC}(\alpha_{r-3} \oplus k_2))\\ &= \mathrm{ISR}(\mathrm{IMC}(a_{r-3})) \oplus \mathrm{ISR}(\mathrm{IMC}(k_2))\\ &= \alpha'_{r-3} \oplus k'_2,\\ \hat{\beta}_{r-4} &= \mathrm{ISR}(\mathrm{IMC}(\hat{\alpha}_{r-3} \oplus k_2))\\ &= \mathrm{ISR}(\mathrm{IMC}(\hat{\alpha}_{r-3})) \oplus \mathrm{ISR}(\mathrm{IMC}(k_2))\\ &= \hat{\alpha}'_{r-3} \oplus k'_2,\end{aligned}$$

where

$$\alpha'_{r-3} = \mathrm{ISR}(\mathrm{IMC}(\alpha_{r-3})), \quad \hat{\alpha}'_{r-3} = \mathrm{ISR}(\mathrm{IMC}(\hat{\alpha}_{r-3})), \quad k'_2 = \mathrm{ISR}(\mathrm{IMC}(k_2)).$$

Hence,

$$\begin{aligned}\delta_{r-6} \oplus \hat{\delta}_{r-6} =&\mathrm{IAC}(\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{IAC}(\mathrm{ISC}(\beta_{r-4})))))) \oplus \mathrm{IAC}(\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{IAC}(\mathrm{ISC}(\hat{\beta}_{r-4}))))))\\ =&\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{AC}(\mathrm{ISC}(\alpha'_{r-3} \oplus k'_2))))) \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}(\mathrm{AC}(\mathrm{ISC}(\hat{\alpha}'_{r-3} \oplus k'_2)))).\end{aligned}$$

And

$$\begin{cases}(\delta_{r-6} \oplus \hat{\delta}_{r-6})^{4i} = \varphi_\eta^{4i},\\ (\delta_{r-6} \oplus \hat{\delta}_{r-6})^{4i+1} = \varphi_\eta^{(4i+13) \bmod 16},\\ (\delta_{r-6} \oplus \hat{\delta}_{r-6})^{4i+2} = \varphi_\eta^{(4i+10) \bmod 16},\\ (\delta_{r-6} \oplus \hat{\delta}_{r-6})^{4i+3} = \varphi_\eta^{(4i+7) \bmod 16},\end{cases}$$

where φ_η denotes all possible solutions of $(\gamma_{r-6} \oplus \hat{\gamma}_{r-6})^j \neq 0$, $0 \leqslant \eta \leqslant 15^4$–1, $0 \leqslant i \leqslant 3$, and $0 \leqslant j \leqslant 15$. Hence,

$$\begin{cases}\mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_0(\mathrm{AC}(\mathrm{ISC}({\alpha'}^{4i}_{r-3} \oplus k'^{4i}_2)))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_0(\mathrm{AC}(\mathrm{ISC}(\hat{\alpha}'^{4i}_{r-3} \oplus k'^{4i}_2))))) = \varphi_\eta^{4i},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_1(\mathrm{AC}(\mathrm{ISC}({\alpha'}^{4i+1}_{r-3} \oplus k'^{4i+1}_2)))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_1(\mathrm{AC}(\mathrm{ISC}(\hat{\alpha}'^{4i+1}_{r-3} \oplus k'^{4i+1}_2))))) = \varphi_\eta^{(4i+13) \bmod 16},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_2(\mathrm{AC}(\mathrm{ISC}({\alpha'}^{4i+2}_{r-3} \oplus k'^{4i+2}_2)))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_2(\mathrm{AC}(\mathrm{ISC}(\hat{\alpha}'^{4i+2}_{r-3} \oplus k'^{4i+2}_2))))) = \varphi_\eta^{(4i+10) \bmod 16},\\ \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_3(\mathrm{AC}(\mathrm{ISC}({\alpha'}^{4i+3}_{r-3} \oplus k'^{4i+3}_2)))))\\ \quad \oplus \mathrm{ISC}(\mathrm{ISR}(\mathrm{IMC}|_3(\mathrm{AC}(\mathrm{ISC}(\hat{\alpha}'^{4i+3}_{r-3} \oplus k'^{4i+3}_2))))) = \varphi_\eta^{(4i+7) \bmod 16},\end{cases}$$

where $0 \leqslant i \leqslant 3$. The secret key, K, is deduced as

$$K = k_1 || k_2 = K_1 || \mathrm{MC}(\mathrm{SR}(k'_2)).$$

5 Attacking complexity

5.1 A rectangle view

The previously defined variables can be placed in a kind of rectangle (as in the rectangle attack), where in one dimension, we have the difference between the correct text and faulty text, and in the other dimension, we have the difference between the observed values (computed by the attacked device with the correct key) and predicted values (computed by the attackers with the key guess). We now provide an analysis based on a single column. We know that MixColumnsSerial maps an input difference with only one non-zero nibble, always to an output difference with four non-zero nibbles. There are $4 \cdot 15 = 60$ such nibbles. This is shown in the third row of Table 4. Similarly, for other types of inputs, we count the number of possible inputs in that case and count the number of times they are mapped to an output with 1, 2, 3, or 4 nonzero nibbles (Table 4).

Table 4 The relation between the numbers of nonzero input and output nibbles in MixColumnsSerial

	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	0	60
2	0	0	0	360	990
3	0	0	360	3600	9540
4	0	60	990	9540	40035

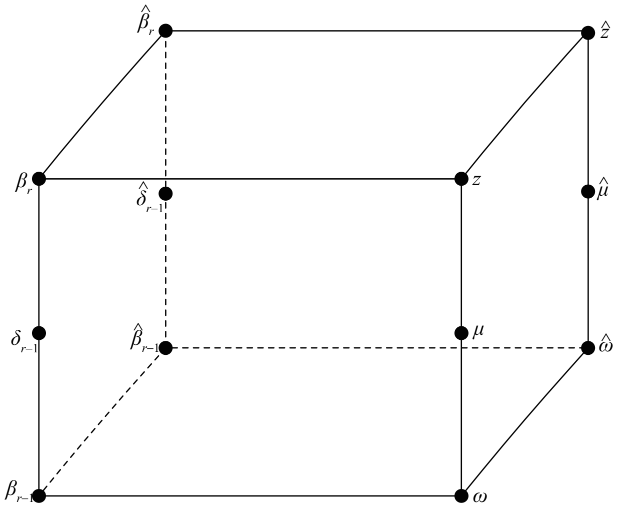

Figure 3 The relationships among variables.

5.2 Computing the probability

Lemma 1. As for the impossible meet-in-the-middle fault analysis on the LED, the probability that a wrong key guess survives a test is 0.774.

Proof. In the proof, we ignore the final linear transformations. There are relations between β_r and z as Figure 3 shows.

$$z = \beta_r \oplus k_1 \oplus g,$$
$$\hat{z} = \hat{\beta}_r \oplus k_1 \oplus g.$$

Assume that $\beta_{r-1} \oplus \bar{\beta}_{r-1}$ takes all 15^4 values without Zeros, equally likely. We compute the probability that a wrong key guess survives a test. The computation is based on the probability of truncated differentials. For the SubCells, the truncated output difference equals the truncated input difference with probability 1. Hence,

$$\beta_r * \hat{\beta}_r = \delta_{r-1} * \hat{\delta}_{r-1},$$

and

$$z * \hat{z} = \mu * \hat{\mu},$$

where $*$ represents the truncated difference. Also, because addition with a subkey does not change the difference, $\beta_r * \hat{\beta}_r = z * \hat{z}$. Furthermore, there are probabilistic relations between $\delta_{r-1} * \hat{\delta}_{r-1}$ and $\beta_{r-1} * \hat{\beta}_{r-1}$, and between $\mu * \hat{\mu}$ and $\omega * \hat{\omega}$, determined by the numbers in Table 4. Finally, we derive that the weight of $\beta_{r-1} * \hat{\beta}_{r-1}$ is always 4 and a wrong key is discarded if the weight of $\omega * \hat{\omega}$ is smaller than 4.

$$\begin{aligned}
&\Pr(\mathrm{wt}(\omega * \hat{\omega}) = 4) \\
&= \sum_{d=1}^{4} (\Pr(d = \mathrm{wt}(\delta_{r-1} * \hat{\delta}_{r-1})) \cdot \Pr(\mathrm{wt}(\omega * \hat{\omega}) = 4 | d = \mathrm{wt}(z * \hat{z}))) \\
&= \sum_{d=1}^{4} (p_1(d) \cdot p_2(d)).
\end{aligned}$$

Table 5 The probability that a wrong key guess survives a test

d	$p_1(d)$	$p_2(d)$	$p_1(d) \cdot p_2(d)$
0	0	0	0
1	60/50625	1	60/50625
2	990/50625	990/1350	98010/68343750
3	9540/50625	9540/13500	91011600/683437500
4	40035/50625	40035/50625	1602801225/2562890625
$\sum$	–	–	0.774

Table 5 shows the values for $p_1(d)$ and $p_2(d)$. They are computed from the entries in Table 3. Thus, we could compute the probability that a wrong key guess survives a test is 0.774.

5.3 Computing the number of faults

Lemma 2. For $q \geqslant 1$,

$$\sigma_q = 2^{16} - 2^{16}(1 - 2^{-2.13})^q,$$

where q represents the number of faults on average, and σ_q denotes the amount of the removed subkey candidates with q faults.

Proof. Because the attackers perform a brute force search on each column with the complexity of 2^{16}, the attackers could remove

$$2^{16} \cdot (1 - 0.774) \approx 2^{13.85},$$

candidates for every column of a subkey by applying one pair of correct and faulty ciphertexts, where the probability of a wrong key guess surviving a test is 0.774 in Lemma 1. When other faults are induced, the subkey space can cover partial candidates of the original subkey space. The overlap of the two groups of equations is computed as

$$\frac{(2^{13.85})^2}{2^{16}} = \frac{2^{27.70}}{2^{16}} = 2^{11.70}.$$

Hence, σ_{q+1} and σ_q have a recursive relationship as

$$\sigma_{q+1} = 2^{13.85} + \sigma_q(1 - 2^{-2.15}),$$

where $q \geqslant 1$ and $\sigma_0 = 0$. The attackers can solve the above recursive formula and derive

$$\sigma_q = 2^{16} - 2^{16}(1 - 2^{-2.15})^q.$$

Theorem 1. In an impossible meet-in-the-middle fault attack on the LED, the attackers can recover one subkey by injecting 43.44 faults into the (r– 4)-th round, where $r \in \{32, 48\}$.

Proof. The subkey space decreases

$$\sigma_q = 2^{16} - 2^{16}(1 - 2^{-2.15})^q,$$

from the above Lemma 1 and 2, if the attackers use q equations. The space of the secret key candidates must be 1 and hold

$$\sigma_q = 2^{16} - 1.$$

That is,

$$q = \frac{-16\log(2)}{\log(1 - 2^{-2.15})} \approx 43.44.$$

Hence, breaking LED-64 and LED-128 require 43.44 and 86.88 faults on average, respectively.

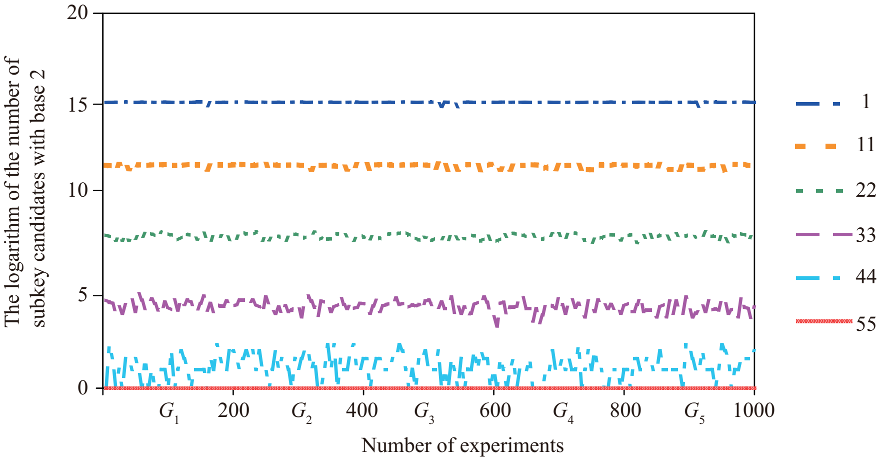

Figure 4 (Color online) The intersections of the subkey candidates in 1000 experiments.

5.4 Computing the complexity

The attacker can perform a brute-force search for one fault injection with the time complexity of

$$4 \cdot 2^{16} \cdot 15^4 \approx 2^{33.63}.$$

The time complexity to break the LED is

$$15^4 + \theta \cdot 2^{33.63},$$

where θ denotes the number of faults. The value of θ is 43.44 for LED-64 and 86.88 for LED-128, respectively. Hence, to break LED-64 in theory, the data and time complexity are 43.44 chosen plaintext-ciphertext pairs on average, and

$$15^4 + 43.44 \cdot 2^{33.63} \approx 2^{39.07},$$

respectively. To break LED-128 in theory, the data and time complexity are 86.88 chosen plaintext-ciphertext pairs, and

$$15^4 + 86.88 \cdot 2^{33.63} \approx 2^{40.07},$$

respectively.

6 Simulation

The attack environment includes three servers with 32-core processors and 64 GB memory using Java. The fault injections are simulated with 1000 process units by computer software. Accuracy, reliability, and latency are taken into consideration for evaluating the experimental results. Figure 4 illustrates the intersections of the subkey candidates, where the x-coordinate and y-coordinate denote the number of evaluated experiments and logarithm of the subkey candidates with base 2, respectively. The colored lines reflect the trend of the 1st, 11th, 22nd, 33rd, 44th, and 55th intersections, respectively.

The accuracy illustrates how close the subkey candidates are to the true subkey. If the number of subkey candidates is close to one, the simulation is regarded as more accurate. The root mean-square error (RMSE) is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{e=1}^{n}(h_e - 1)},$$

where n denotes the number of experiments in a subset, e represents the index of each experiment, and h_e denotes the number of subkey candidates. The RMSE trend for each intersection of the subkey candidates is shown in Table 6, where $n = 200$ and $e \in \{1, \ldots, 1000\}$. Further, all experiments are categorized into

Table 6 The subkey recovery on accuracy by RMSE

Group	1st intersection	11th intersection	22nd intersection	33rd intersection	44th intersection	55th intersection
G_1	197.33	61.09	16.59	4.47	1.19	0
G_2	197.30	60.72	16.61	4.47	1.18	0
G_3	197.28	60.97	16.63	4.46	1.18	0
G_4	197.10	61.07	16.64	4.49	1.21	0
G_5	197.17	61.12	16.61	4.47	1.16	0

Table 7 The subkey recovery on reliability

Group	1st intersection	11th intersection	22nd intersection	33rd intersection	44th intersection	55th intersection
G_1	0%	0%	0%	0%	22.5%	100%
G_2	0%	0%	0%	0%	25.0%	100%
G_3	0%	0%	0%	0%	24.5%	100%
G_4	0%	0%	0%	0%	22.0%	100%
G_5	0%	0%	0%	0%	22.0%	100%

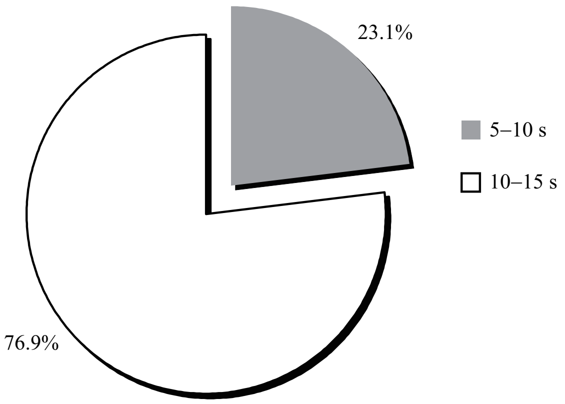

Figure 5 The subkey recovery on latency.

five groups on average, denoted as G_1, G_2, G_3, G_4, and G_5. This illustrates that the accuracy in each group for the same interaction is appropriate.

Reliability describes the success rate in all experiments. The attack is regarded as successful until the attackers can derive only one subkey. The success rates on average are 0%, 0%, 0%, 0%, 23.2%, and 100% in Table 7. The attackers had to inject 44.20 random faults on average to derive one subkey. To break LED-64 and LED-128, the data complexities are 44.20 and 88.40 chosen plaintext-ciphertext pairs on average, and the time complexities are

$$15^4 + 44.20 \cdot 2^{33.63} \approx 2^{39.10},$$

and

$$15^4 + 88.40 \cdot 2^{33.63} \approx 2^{40.10},$$

respectively.

Latency is the time of recovery for one subkey. The latency of all experiments is between 5 and 15 s in Figure 5.

7 Conclusion

This paper proposes a novel impossible meet-in-the-middle fault attack on the LED in a nibble-oriented fault model. The IMFA attack could break LED-64 and LED-128 with only 44.20 and 88.40 faults on average, respectively. The attackers can provoke faults into the deeper rounds of the LED by x-ray, radiation, or micro-probe in the hardware implementation, or alter the internal state of the code in the vehicle device software implementation. Hence, it is suggested that the first and last five rounds of the LED be protected from fault analysis in VANETs.

Acknowledgements This work was supported by Research Council KU Leuven (Grant No. OT/13/071), National Key Basic Research Program of China (Grant No. 2013CB338004), National Natural Science Foundation of China (Grant Nos. 61772129, 61472250, 61402286, 61672347, 61402288), Innovation Program of Shanghai Municipal Education Commission (Grant No. 14ZZ066), Shanghai Natural Science Foundation (Grant Nos. 15ZR1400300, 16ZR1401100), European Union's Horizon 2020 Research and Innovation Programme (Grant No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET), Open Research Fund of State Key Laboratory of Information Security (Grant No. AGK20170X), National Cryptography Development Fund (Grant No. MMJJ20170214), Fundamental Research Funds for the Central Universities, and China Scholarship Council (Grant No. CSC201403170380).

References

1 Misener A J. Vehicle-infrastructure integration (VII) and safety: rubber and radio meets the road in California. Intellimotion, 2005, 11: 1–12

2 Hubaux P J, Capkun S, Luo J. The security and privacy of smart vehicles. IEEE Secur Priv, 2004, 2: 49–55

3 Raya M, Hubaux P J. Securing vehicular ad hoc networks. J Com Secur, 2007, 15: 39–68

4 Raya M, Papadimitratos P, Hubaux P J. Securing vehicular communications. IEEE Trans Dependable Secure Comput, 2006, 13: 8–15

5 Zhang W T, Bao Z Z, Lin D D, et al. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci China Inf Sci, 2015, 58: 122103

6 Li L, Liu B T, Wang H. QTL: a new ultra-lightweight block cipher. Microprocessor Microsy, 2016, 45: 45–55

7 Engels D, Saarinen O J M, Schweitzer P, et al. The Hummingbird-2 lightweight authenticated encryption algorithm. In: Proceedings of the 7th International Conference on RFID Security and Privacy, Amherst, 2011. 19–31

8 Hong D, Sung J, Hong S, et al. HIGHT: a new block cipher suitable for low-resource device. In: Proceedings of the 8th International Conference on Cryptographic Hardware and Embedded Systems, Yokohama, 2006. 46–59

9 Lim H C, Korkishko T. mCrypton-a lightweight block cipher for security of low-cost RFID tags and sensors. In: Proceedings of the 6th International Conference on Information Security Applications, Jeju Island, 2005. 243–258

10 Ojha K S, Kumar N, Jain K. TWIS-a lightweight block cipher. In: Proceedings of the 5th International Conference on Information Systems Security, Kolkata, 2009. 280–291

11 Bogdanov A, Knudsen L R, Lender G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, 2007. 450–466

12 Wu W L, Zhang L. LBlock: a lightweight block cipher. In: Proceedings of the 9th International Conference on Applied Cryptography and Network Security, Nerja, 2011. 327–344

13 Dai X, Huang Y, Chen L, et al. VH: a lightweight block cipher based on dual pseudo-random transformation. In: Proceedings of International Conference on Cloud Computing and Security, Nanjing, 2015. 3–13

14 Guo J, Peyrin T, Poschmann A, et al. The LED block cipher. In: Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems, Nara, 2011. 326–341

15 Mendel F, Rijmen V, Toz D, et al. Differential analysis of the LED block cipher. In: Proceedings of the 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, 2012. 190–207

16 Isobe T, Shibutani K. Security analysis of the lightweight block ciphers XTEA, LED and Piccolo. In: Proceedings of the 17th Australasian Conference on Information Security and Privacy, Wollongong, 2012. 71–86

17 Nikolić I, Wang L, Wu S. Cryptanalysis of round-reduced LED. In: Proceedings of International Workshop on Fast Software Encryption, Washington, 2013. 112–129

18 Soleimany H. Probabilistic slide cryptanalysis and its applications to LED-64 and Zorro. In: Proceedings of International Workshop on Fast Software Encryption, London, 2014. 373–389

19 Jeong K, Lee C. Differential fault analysis on block cipher LED-64. In: Future Information Technology, Application, and Service. Berlin: Springer, 2012. 747–775

20 Li W, Gu D W, Xia X L, et al. Single byte differential fault analysis on the LED lightweight cipher in the wireless sensor network. Int J Comput Intell Syst, 2012, 5: 896–904

21 Jovanovic P, Kreuzer M, Polian I. A fault attack on the LED block cipher. In: Proceedings of the 3rd International Conference on Constructive Side-Channel Analysis and Secure Design, Darmstadt, 2012. 120–134

22 Zhao X J, Guo S Z, Zhang F. Improving and evaluating differential fault analysis on LED with algebraic techniques. In: Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Washington, 2013. 41–51

23 Ghalaty F N, Yuce B, Schaumont P. Differential fault intensity analysis on PRESENT and LED block ciphers. In: Proceedings of the 6th International Workshop on Constructive Side-Channel Analysis and Secure, Berlin, 2015. 174–188

24 Li W, Zhang W W, Gu D W, et al. Impossible differential fault analysis on the LED lightweight cryptosystem in the vehicular ad-hoc networks. IEEE Trans Depend Secure Comput, 2016, 13: 84–92

25 Boneh D, DeMillo A R, Lipton J R. On the importance of eliminating errors in cryptgraphic computations. J Cryptol, 2001, 14: 101–119

26 Boneh D, DeMillo A R, Lipto J R, et al. On the importance of checking cryptographic protocols for faults. In: Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques, Konstanz, 1997. 37–51

27 Dusart P, Letourneux G, Vivolo O. Differential fault analysis on A.E.S. In: Proceedings of International Conference on Applied Cryptography and Network Security, Kunming, 2003. 293–306

28 Blömer J, Seifert J P. Fault based cryptanalysis of the advanced encryption standard (AES). In: Proceedings of International Conference of Financial Cryptography, Guadeloupe, 2003. 162–181

29 Zhang F, Zhao X J, He W, et al. Low-cost design of stealthy hardware trojan for bit-level fault attacks on block ciphers. Sci China Inf Sci, 2017, 60: 048102

30 Zhao X J, Zhang F, Guo S Z, et al. Optimal model search for hardware-trojan-based bit-level fault attacks on block ciphers. Sci China Inf Sci, 2018, 61: 039106

31 Liao N, Cui X X, Liao K, et al. Improving DFA attacks on AES with unknown and random faults. Sci China Inf Sci, 2017, 60: 042401

32 Derbez P, Fouque A P, Lereateux D. Meet-in-the-middle and impossible differential fault analysis on AES. In: Proceedings of International Workshop of Cryptographic Hardware and Embedded Systems, Nara, 2011. 274–291