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

Li, Wei; Rijmen, Vincent; Tao, Zhi; Wang, Qingju; Chen, Hua; Liu, Yunwen; Li, Chaoyun; Liu, Ya

Published in:
Science China Information Sciences

Link to article, DOI:
10.1007/s11432-017-9209-0

Publication date:
2018

Document Version
Peer reviewed version

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], and 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
liuyaloccs@gmail.com

Abstract. With the enlargement of wireless technology, vehicular ad-hoc networks (VANETs) are emerging as a promising approach to realizing smart cities and addressing lots of serious traffic problems such as road safety, convenience and efficiency. In order to avoid any possible rancorous attacks, employing lightweight ciphers is most effective to implement encryption/decryption, message authentication and digital signature for security of VANETs. LED is a lightweight block cipher with two basic keysize variants, including LED-64 and LED-128. Since its designing, a multitude of fault analysis techniques focus on provoking faults at the last four rounds of LED to derive the 64-bit and 128-bit secret keys. It is vital to investigate whether injecting faults into a former round allows breaking LED. This study presents a novel impossible meet-in-the-middle fault analysis on one round earlier of LED. A detailed analysis of the expected number of faults is proved to uniquely determine the secret key. It is based on the propagation of truncated differentials and 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 LED by fault injections.

⋆ Corresponding author

Keywords:VANETs, LED, Lightweight cipher, Impossible meet-in-the-middle, Fault analysis

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. It has been tremendously successful and naturally attracted considerable attention from both academia and industry [1]. However, VANETs are networks with high dynamic topology and their connections is vulnerable to attacks. For instance, attackers may exploit VANETs to send bogus information to deceive other vehicles. Therefore, conservation of security in VANETs is an indispensable demand. Nodes in VANETs should be confident that each communication has been started from a trustworthy source node and messages are not varied by malicious vehicles. Although these issues seem similar to those used in traditional communication networks, there are individual characteristic for VANETs. The seriousness of security failures, the selforganized nature of network, the high mobility of vehicles, the relevance of vehicles to their geographic position, and the 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 lots of security applications, such as encryption/decryption, message authentication, and digital signature, etc. It is very serious and urgent to implement effective ciphers in VANETs, i.e., lightweight ciphers are mostly selected for confidentiality, authentication and integrity [5-13]. Thus, appliance of lightweight ciphers can reduce energy consumption for devices, and allow more network communications with lower-resource devices in vehicles.

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

In the last two decades, fault analysis puts forward a serious threat for cryptographic implementation. It can deduce the secret key by applying the mathematical relations of a cipher resulting from correct and faulty operations. Boneh et al. presented RSA against fault analysis by provoking the faulty bits in 1996 [25, 26]. 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) et al., were later proposed to break block ciphers [27-29]. The attackers can inject faults into the running procedure by exploring a glitch on the clock, a spike on the power supply, or implementing the external ways of the laser and electromagnetic radiations. They makes 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 about the secret key by examining the differential, algebraic, statistical or impossible differential relations to recover the subkeys, respectively. Table 1 illustrates the comparison of the latest fault analysis results on LED. Three research groups proposed DFA to break LED in the same year [19-21]. They can break the last subkey by injecting faults into the antepenultimate round of LED. Jeong et al. can derive the 64-bit secret key by one random nibble fault injection. Li et al. extended a random nibble-oriented fault model to a random byte-oriented fault model, and break LED-64 and LED-128 with 3 and 6 faults, respectively. Jovanovic et al. applied some techniques of proportional relationships between different layers to reduce the number of faults to 1 and 2, respectively. Then Zhao et al. proposed an algebraic fault analysis (AFA) by inducing the same faults into the antepenultimate round [22]. They used an algebraic relationship to describe the intermediate value of LED, respectively. Depending on on the statistical relationship, Ghalaty et al. 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 [23]. In 2016, Li et al. presented an IDFA on LED and extended fault locations to the third last round with 48 and 96 faults, respectively [24]. Hence, the previous fault analysis only targets on the last four rounds of LED.

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

Adding protection to full rounds of a lightweight cipher is ideal against fault attack for the high-constrained devices in VANETs. However, it can decrease the performance, and are 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 the real applications, random faults can occur in any round or register of the lightweight cipher. It is excellent that the high-constrained devices can be resistant 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. It is the motivation why we investigate a novel fault analysis by attacking earlier rounds of LED.

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

The remainder of this paper is organized as follows. Section 2 describes the specification of 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 to break LED-64 and LED-128. The next two sections presented the attacking complexity and analyze the experimental results. The last section concludes the paper.

2 Specification of LED

The LED lightweight cipher fixes the block lengh to 64 bits, and supports key lengths of 64 bits and 128 bits [14]. It has 32 and 48 rounds for LED-64 and LED-128 as Fig. 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.
- ShiftRows(SR) cyclically shifts each row of the state by different offsets.
- MixColumnsSerial(MC) takes all the columns and multiply their data with a matrix.

The sequence of steps for the decryption is same as that for 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 LED as Table 2 shows.

3 The IDFA and MFA attack on LED

3.1 Notations

The notations of LED and its analysis are described as Table 3 shows:

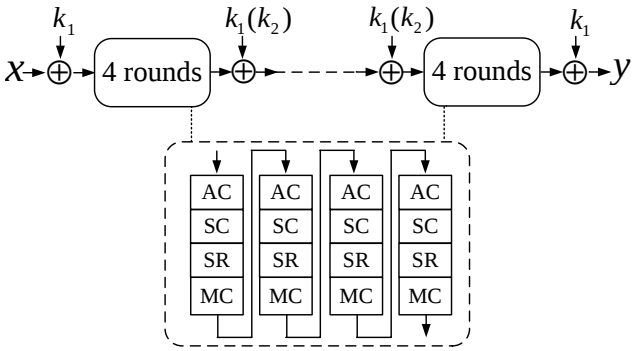

Fig. 1. Structure of LED.

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

Notions	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 $1 \leq l \leq r$
$\hat{\alpha}_l$, $\hat{\beta}_l$, $\hat{\gamma}_l$, $\hat{\delta}_l$	the 64-bit faulty output of the AC, SC, SR and MC layers in the l-th round with $1 \leq l \leq r$
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

3.2 fault model and main procedure

The fault model includes chosen plaintext attacks and random nibble-oriented fault model. The IDFA and MFA are two independent different kinds of fault analysis, which are proposed to attack AES [29]. Some random faults are injected into the third last round of the running procedure, and thus right 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 $0 \leq i \leq 3$. 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 $0 \leq i \leq 3$. 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 drived. In [24], the IDFA attack can recover LED-64 and LED-128 with 48 and 96 faults, respectively. There is no experimental results about the MFA attack on 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 at the (r–4)th round, where $r \in \{32, 48\}$. As Fig. 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

$$\begin{aligned} \beta_r &= ISR(IMC(y \oplus k_1)) \\ &= ISR(IMC(y)) \oplus ISR(IMC(k_1)) \\ &= y' \oplus k'_1, \end{aligned}$$

$$\begin{aligned} \hat{\beta}_r &= ISR(IMC(\hat{y} \oplus k_1)) \\ &= ISR(IMC(\hat{y})) \oplus ISR(IMC(k_1)) \\ &= \hat{y}' \oplus k'_1, \end{aligned}$$

where

$$\begin{aligned} y' &= ISR(IMC(y)), \\ \hat{y}' &= ISR(IMC(\hat{y})), \\ k'_1 &= ISR(IMC(k_1)). \end{aligned}$$

And

$$\begin{aligned}\delta_{r-2} \oplus \hat{\delta}_{r-2} =& IAC(ISC(ISR(IMC(IAC(ISC(\beta_r)))))) \oplus \\ & IAC(ISC(ISR(IMC(IAC(ISC(\hat{\beta}_r)))))) \\ =& ISC(ISR(IMC(AC(ISC(y' \oplus k'_1))))) \oplus \\ & ISC(ISR(IMC(AC(ISC(\hat{y}' \oplus k'_1))))).\end{aligned}$$

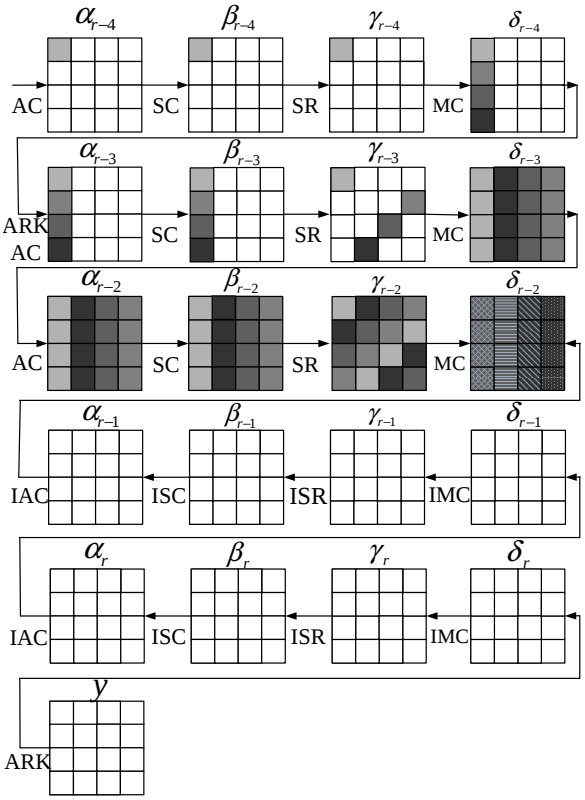

Fig. 2. One of the fault attacking paths in the last five rounds.

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

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

where $0 \leq j \leq 15$. 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:

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

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

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

where $0 \leq i \leq 3$. The above equations allow to restrict possible candidates for k_1'. The attackers can do 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:

$$K = k_1 = MC(SR(k_1')).$$

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 attacking 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

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

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

where

$$\begin{aligned} \alpha'_{r-3} &= ISR(IMC(\alpha_{r-3})), \\ \hat{\alpha}'_{r-3} &= ISR(IMC(\hat{\alpha}_{r-3})), \\ k'_2 &= ISR(IMC(k_2)). \end{aligned}$$

Hence,

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

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 $(\gamma_{r-6} \oplus \hat{\gamma}_{r-6})^j \neq 0$, $0 \leq \eta \leq 15^4$–1, $0 \leq i \leq 3$, and $0 \leq j \leq 15$. Hence,

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

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

$$K = k_1 || k_2 = K_1 || MC(SR(k'_2)).$$

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 texts and the faulty texts, and in the other dimension we have the difference between the observed values (computed by the attacked device with the correct key) and the predicted values (computed by the attackers with the guess for the key). 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, cf. 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

5.2 Computing the probability

Lemma 1. As for the impossible meet-in-the-middle fault analysis on LED, the probability of 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 Fig. 3 shows:

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

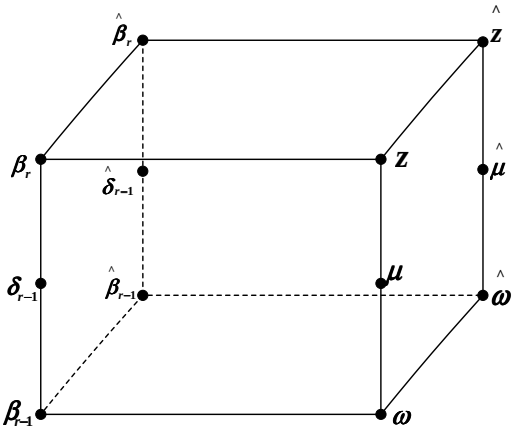

Fig. 3. The relationships among variables.

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.

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

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.

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

d	p_1	p_2	$\cdot$
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

5.3 Computing the number of faults

Lemma 2. For $q \geq 1$,

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

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

Proof. Since the attackers do 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 survives a test

equals 0.774 in Lemma 1. When other faults are induced, the subkey space can cover partial candidates of the original subkey space. The overlap part of 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 the recursive relationship as

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

where $q \geq 1$ 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. In an impossible meet-in-the-middle fault attack on LED, the attackers can recover one subkey by injecting 43.44 faults into the r–4 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 uses q equations. The space of the secret key candidates must be 1 and hold

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

That is,

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

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

5.4 Computing the complexity

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

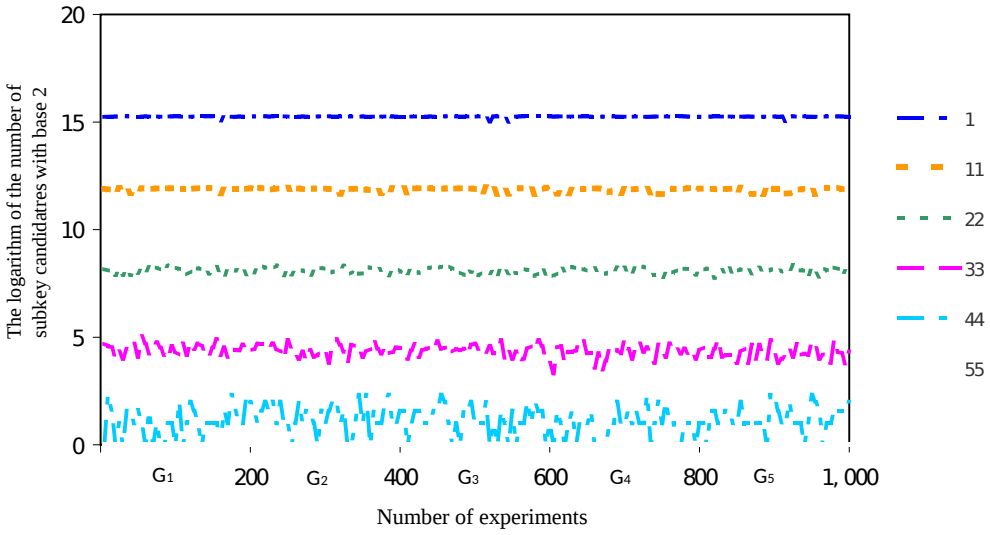

Fig. 4. The intersections of the subkey candidates in 1000 experiments.

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 to be more accurate. The Root Mean-Square Error(RMSE) is defined as

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

where n denotes the number of experiments in a subset, e represents the index of each experiment, h_e denotes the number of subkey candidates. The RMSE trend for every intersection of subkey candidates are shown in Table 6, where $n = 200$ and $e \in \{1, \cdots, 1000\}$. And all experiments are categorized into five groups on average, denoted as G_1, G_2, G_3, G_4 and G_5. It illustrates that accuracy in each group for the same interaction is appropriate.

Table 6. The subkey recovery on Accuracy by RMSE

Groups	1	11	22	33	44	55
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

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, respectively. 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 in average, and the time complexities are

Table 7. The subkey recovery on Reliability

Groups	1	11	22	33	44	55
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%

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

and

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

respectively.

Latency is the time of recovering of one subkey. The latency of 100% experiments is between 5s and 15s in Fig. 5.

7 Conclusions

This paper proposes a novel impossible meet-in-the-middle fault attack on 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 LED by X-Ray, radiation, Micro-Probe in the hardware implementation, or alter the internal state of the code in the software implementation of devices in vehicles. Hence, the first and last five rounds of LED are suggested to be protected from fault analysis in VANETs.

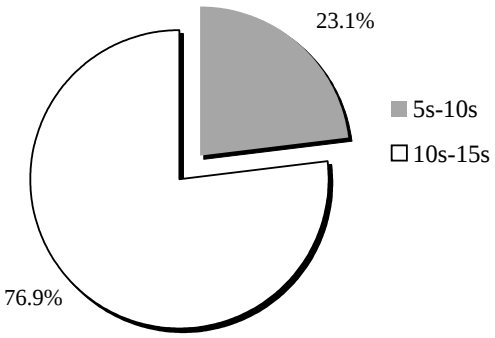

Fig. 5. The subkey recovery on Latency.

Acknowledgments

This work was supported by the 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 No. 61472250, No. 61772271, No. 61402286, No. 61672347, No. 61402288), Innovation Program of Shanghai Municipal Education Commission (Grant No. 14ZZ066), Shanghai Natural Science Foundation (Grant No. 15ZR1400300, No. 16ZR1401100), European Union's Horizon 2020 research and innovation programme (Grant No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET), the open research fund of State Key Laboratory of Information Security (Grant No. AGK20170X), National Cryptography Development Fund (Grant No. MMJJ20170214), the 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(2): 1-12
2. Hubaux P J, Capkun S, Luo J. The security and privacy of smart vehicles. IEEE Secur Priv, 2004, 2(3): 49-55
3. Raya M, Hubaux P J. Securing vehicular ad hoc networks. J Com Secur, 2007, 15(1): 39-68
4. Raya M, Papadimitratos P, Hubaux P J. Securing vehicular communications. IEEE Trans Dependable Secure Comput, 2006, 13(5): 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, 2014, 58(12): 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: Juels A, Paar C, eds. International Workshop on Radio Frequency Identification: Security and Privacy Issues. Lect Notes in Comp Sci, vol 7055. Berlin: Springer-Verlag, 2011. 19-31

8. Hong D, Sung J, Hong S, et al. HIGHT: a new block cipher suitable for low-resource device. In: Goubin L, Matsui M, eds. International Workshop on Cryptographic Hardware and Embedded Systems. Lect Notes in Comp Sci, vol 4249. Berlin: Springer-Verlag, 2006. 46-59
9. Lim H C, Korkishko T. mCrypton-a lightweight block cipher for security of low-cost RFID tags and sensors. In: Song J-S, Kwon T, Yung M, eds. International Workshop on Information Security Applications. Lect Notes in Comp Sci, vol 3786. Berlin: Springer-Verlag, 2005. 243-258
10. Ojha K S, Kumar N, Jain K. TWIS-a lightweight block cipher. In: Prakash A, Gupta S I, eds. International Conference on Information Systems Security. Lect Notes in Comp Sci, vol 5905. Berlin: Springer-Verlag, 2009. 280-291
11. Bogdanov A, Knudsen R H, Lender G, et al. PRESENT: an ultra-lightweight block cipher. In: Paillier P, Verbauwhede I, eds. International Workshop on Cryptographic Hardware and Embedded Systems. Lect Notes in Comp Sci, vol 4727. Berlin: Springer-Verlag, 2007. 450-466
12. Wu W L, Zhang L. LBlock: a lightweight block cipher. In: Lopez J, Tsudik G, eds. International Conference on Applied Cryptography and Network Security. Lect Notes in Comp Sci, vol 6715. Berlin: Springer-Verlag, 2011. 327-344
13. Dai X, Huang Y, Chen L, et al. VH: a lightweight block cipher based on dual pseudo-random transformation. In: Huang Z Q, Sun X M, Luo J Z, eds. International Conference on Cloud Computing and Security. Lect Notes in Comp Sci, vol 9483. Berlin: Springer-Verlag, 2015. 3-13
14. Guo J, Peyrin T, Poschmann A, et al. The LED block cipher. In: Preneel B, Takagi T, eds. International Workshop of Cryptographic Hardware and Embedded Systems. Lect Notes in Comp Sci, vol 6917. Berlin: Springer-Verlag, 2011. 326-341
15. Mendel F, Rijmen V, Toz D, et al. Differential analysis of the LED block cipher. In: Wang X Y, Sako K, eds. International Conference on the Theory and Application of Cryptology and Information Security. Lect Notes in Comp Sci, vol 7658. Berlin: Springer-Verlag, 2012. 190-207
16. Isobe T, Shibutani K. Security analysis of the lightweight block ciphers XTEA, LED and Piccolo. In: Susilo W, Mu Y, Seberry J, eds. Australasian Conference of Information Security and Privacy. Lect Notes in Comp Sci, vol 7372. Berlin: Springer-Verlag, 2012. 71-86
17. Nikolić I, Wang L, Wu S. Cryptanalysis of round-reduced LED. In: Moriai, eds. International Workshop of Fast Software Encryption. Lect Notes in Comp Sci, vol 8424. Berlin: Springer-Verlag, 2013. 112-129
18. Soleimany H. Probabilistic slide cryptanalysis and its applications to LED-64 and Zorro. In: Cid C, Rechberger C, eds. International Workshop of Fast Software Encryption. Lect Notes in Comp Sci, vol 8540. Berlin: Springer-Verlag, 2014. 373-389.
19. Jeong K, Lee C. Differential fault analysis on block cipher LED-64. In: Park J J, Leung C M V, Wang C L, et al., eds. Future Information Technology, Application, and Service. Lect Notes in Electr Eng, vol 164. 2012, 164: 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 Comp Intell Sys, 2012, 5(5): 896-904
21. Jovanovic P, Kreuzer M, Polian I. A fault attack on the LED block cipher. In: Schindler W, Huss A S, eds. International Workshop of Constructive Side-Channel Analysis and Secure Design, Lect Notes in Comp Sci, vol 7275. Berlin: Springer-Verlag, 2012. 120-134

22. Zhao X J, Guo S Z, Zhang F. Improving and evaluating differential fault analysis on LED with algebraic techniques. In: Fischer W, Schmidt J M, eds. International Workshop on Fault Diagnosis and Tolerance in Cryptography, IEEE Computer Society. 2013. 41-51
23. Ghalaty F N, Yuce B, Schaumont P. Differential fault intensity analysis on PRESENT and LED block ciphers. In: Mangard S, Poschmann Y A, eds. International Workshop of Constructive Side-Channel Analysis and Secure Design. Lect Notes in Comp Sci, vol 9064. Berlin: Springer-Verlag, 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 Dependable Secure Comput, 2016, 13(1): 84-92
25. Boneh D, DeMillo A R, Lipton J R. On the importance of eliminating errors in cryptgraphic computations. J Cryptol, 2001, 14(2): 101-119
26. Boneh D, DeMillo A R, Lipto J R, et al. On the importance of checking cryptographic protocols for faults. In: Fumy W, eds. International Conference on the Theory and Application of Cryptographic Techniques. Lect Notes in Comp Sci, vol 1233. Berlin: Springer-Verlag, 1997. 37-51
27. Dusart P, Letourneux G, Vivolo O. Differential fault analysis on A.E.S. In: Zhou J Y, Yung M, Han Y F, eds. International Conference of Applied Cryptography and Network Security. Lect Notes in Comp Sci, vol 2846. Berlin: Springer-Verlag. 2003. 293-306
28. Blömer J, Seifert J P. Fault based cryptanalysis of the advanced encryption standard (AES). In: Wright N. R. eds. International Conference of Financial Cryptography. Lect Notes in Comp Sci, vol 2743. Berlin: Springer-Verlag, 2003. 162-181
29. Derbez P, Fouque A P, Lereateux D, Meet-in-the-middle and impossible differential fault analysis on AES. In: Preneel B, Takagi T, eds. International Workshop of Cryptographic Hardware and Embedded Systems, Lect Notes in Comp Sci, vol 6917. Berlin: Springer-Verlag, 2011. 274-291