Improved See-In-The-Middle Attacks on AES*

Jonghyun Park[1], Hangi Kim[1], and Jongsung Kim[1,2]

[1] Department of Financial Information Security, Kookmin University, Republic of Korea {mmo330,tiontta,jskim}@kookmin.ac.kr

[2] Department of Information Security, Cryptology, and Mathematics, Kookmin University, Republic of Korea

Abstract. The See-In-The-Middle attack is designed to work effectively even with a low signal to noise ratio; hence, it can be performed even with poor side-channel analysis tools. Because it exploits the side-channel leakage of the middle round of the block cipher implementations, it is effective for implementations with reduced masking. In this study, we propose attacks to improve the See-In-The-Middle attack against the 4-round masked implemented AES introduced in the previous work. In addition, we present an attack against AES-256 implemented with 12-round reduced masking to recover 2-byte of the master key using the related-key differential trail, showing that the See-In-The-Middle attack is only thwarted by masking the whole rounds of AES-256 in the related-key model.

Keywords: AES · Side-channel analysis· SITM · Middle rounds attack · Differential cryptanalysis

1 Introduction

The side-channel analysis (SCA) proposed in 1996 is currently the most powerful attack technique among the attacks on cryptographic implementations [7]. Many methods that utilize various side-channel information, such as power consumption and electromagnetic emanation of the device for attack, have been proposed [2, 8].

Applying masking to the whole round of the block cipher can be a general countermeasure for all SCAs. However, in practice, it is often applied only to the first and last few rounds of the block cipher because the masking implementation causes a large overhead [10]. The side-channel assisted differential cryptanalyses have been recently proposed. [3, 4, 6]. Among them, S. Bhasin et al. [3] presented the See-In-The-Middle (SITM) that attack targets the reduced masked implementations of block ciphers such as AES, SKINNY and PRESENT. They showed that SITM attack is possible even in harsh experimental environments with a low signal to noise ratio (SNR).

* This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00520, Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes).

Contributions

In this study, we improve the SITM attacks against AES with a 4-round masked implementation. In addition, we show that SITM attacks against AES-256 are also possible in the related-key model. Our attack in the related-key model works on 12-round masked AES-256 and can recover 2-byte of a secret master key. Therefore, full-round masking should be applied to the AES-256 implementation to guarantee security against SCA in the related-key model. The attack complexities are summerized in Table 1. Target depth refers to the number of rounds for measuring power traces through side-channel observation.

Table 1. Comparison of the SITM attack complexities on AES

Distinguisher	Key size	Target depth	Data (chosen PTs)	Memory (bytes)	Time	Ref.
Single-key characteristics	128	3	$2^{13.73}$	2^{10}	$\mathcal{O}(2^{11.5})$	[3]
		3	$2^{7.32}$	2^{11}	$2^{7.32}$	Section 3
	192	3, 4	$2^{14.73}$	2^{10}	$\mathcal{O}(2^{11.5})$	[3]
		3, 4	$2^{8.32}$	2^{11}	$2^{8.32}$	Section 3
	256	3, 4	$2^{14.73}$	2^{10}	$\mathcal{O}(2^{11.5})$	[3]
		3, 4	$2^{8.32}$	2^{11}	$2^{8.32}$	Section 3
Related-key characteristics	256	7	2^{31}	2^{5}	2^{32}	Section 4*

* 2-byte master key recovery, PT: PlainText

2 Attack Methodology

2.1 The Block Cipher AES

The block cipher AES encrypts a 16-byte plaintext using a 128-, 192- and 256-bit master key (MK) and processes 10, 12, and 14 rounds, respectively [9]. For convenience, we labeled the bytes in the cipher state column-wise from left to right:

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

The round function of AES is composed of SubBytes (SB), ShiftRows (SR), MixColumns (MC) and AddRoundKey (AK). SB applies an 8-bit S-box to each byte of the state. SR cyclic left rotates the i^{th} row of the state by i-byte. MC applies a diffusion matrix to each column of the state. AK XORs the r^{th} round key RK_r to the state.

AES-128 uses the master key as the first round key, while AES-192 and AES-256 also use the master key in the first and the second rounds without modification. To help understand the related-key differential trail in Section 4, we will describe the round key generation process of AES-256 below.

MK is divided by 16-byte and used as RK_0 and RK_1. We denote the i^{th} cell of RK_r as $RK_r[i]$. $RK_2 \sim RK_{14}$ are generated by the following process:

$$
\begin{array}{ll}
\multicolumn{2}{l}{\text{For } r = 1, 2, \cdots, 7,} \\
\quad RK_{2r}[i] \leftarrow S(RK_{r+1}[i+13]) \oplus RK_r[i] \oplus Rcon_r, & 0 \leq i \leq 2; \\
\quad RK_{2r}[i] \leftarrow S(RK_{r+1}[12]) \oplus RK_r[3] \oplus Rcon_r, & i = 3; \\
\quad RK_{2r}[i] \leftarrow RK_{2r}[i-4] \oplus RK_r[i], & 4 \leq i \leq 15; \\
\quad RK_{2r+1}[i-16] \leftarrow S(RK_{2r}[i-4]) \oplus RK_{r+1}[i-16], & 16 \leq i \leq 19; \\
\quad RK_{2r+1}[i-16] \leftarrow RK_{2r+1}[i-20] \oplus RK_{r+1}[i-16], & 20 \leq i < 32,
\end{array}
$$

where $S()$ stands for the 8-bit S-box, and $Rcon_r$ is a round dependent constant. Please refer to [9] for more details.

2.2 SITM Overview

The SITM attack is a side-channel assisted differential cryptanalysis that targets reduced masked implementations of a block cipher. Differential cryptanalysis analyzes the difference that changes as the state with difference progresses to the next state. The sequence of the connected states is referred as the differential trail and if the difference is not a specific value other than zero, then it is referred as the differential pattern.

The difference between the side-channel traces occurring during the encryption processes is used for the attack. Suppose that we observe the side-channel leakage occurring in the encryption processes of two different plaintexts. If the S-box is applied to the same values, the two power traces will be similar. However, if the values are different, a recognizable difference will exist between the two power traces. We call the difference between the two power traces as the difference trace. Using this, we can determine whether a pair of encryption processes satisfies the target differential pattern (or differential trail) in the middle round. After finding such encryption pairs, the attacker can deduce the key candidates that can make a valid differential transition. We used the ChipWhisperer-Lite tool for the side-channel observation and implemented AES in C code on ATXMEGA128D4 8-bit RISC [1].

3 Improved SITM Attacks on AES Using Single-Key Differential Patterns

3.1 The Differential Patterns

Our SITM attack uses AES differential patterns other than the ones used in the attack proposed in [3]. We used 32 differential patterns for the attack. Figure 1 presents one of the cases among them.

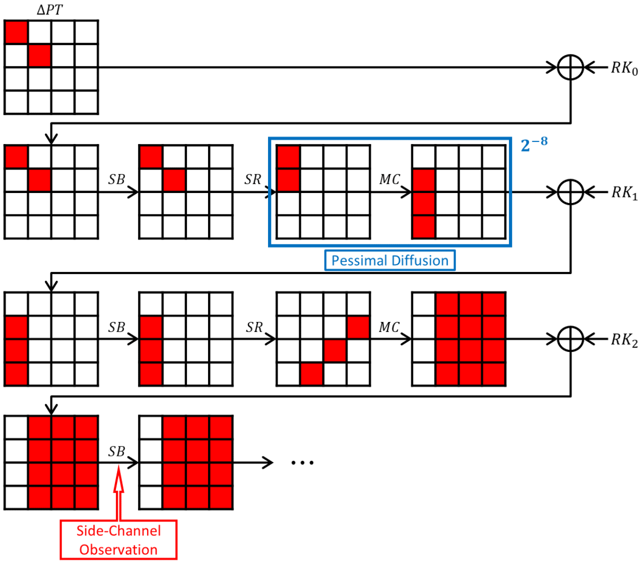

Fig. 1. AES differential pattern (Active cells are colored in red).

In MC, the MDS matrix is applied to each column, such that at least five cells among the input and output cells of the matrix are active.[3] To follow our differential pattern, pessimal diffusion must occur in the first round MC. The number of differential trails, in which pessimal diffusion occurs such that the s_0 cell after the first round is inactive, is $2^8 - 1$; thus, a differential pattern in Figure 1 occurs with a probability of approximately 2^{-8}.

An inactive column is guaranteed in the third round if only one cell among $s_0 \sim s_3$ is inactive after the first round. We call the group of these four differential patterns as a "type". We then define and use the following eight types of differential patterns:

type 1: s_0 and s_5 cells are active in PT.

type 2: s_1 and s_6 cells are active in PT.

[3] A cell with a non-zero difference is called "active".

type 3: s_2 and s_7 cells are active in PT.
type 4: s_3 and s_{14} cells are active in PT.
type 5: s_4 and s_9 cells are active in PT.
type 6: s_8 and s_{13} cells are active in PT.
type 7: s_{10} and s_{15} cells are active in PT.
type 8: s_{11} and s_{12} cells are active in PT.

3.2 Application of SITM

Our SITM attack process is divided into two processes: 1) finding plaintext pairs satisfying the differential pattern; and 2) key-recovery. These processes are independently performed for each type, so that the 2-byte key candidates can be recovered at each execution. This section describes the attack on type 1 as an example, which can be easily transformed into attacks on other types.

Finding plaintext pairs satisfying the differential pattern This process requires the following steps:

1. Randomly generate $2^{4.32}$ plaintexts satisfying the input differential pattern.
2. Encrypt each plaintext and collect the power traces of the third round SB operation.
3. Calculate the difference trace for one of the power trace pairs.
4. Check whether the difference trace has an inactive column in the third round SB operation. Collect the plaintext pair if there is any.
5. Repeat steps 3 and 4 for all difference traces.

A type has four differential patterns; therefore, the probability that a plaintext pair is collected in the abovementioned process is approximately 2^{-6}. We expect at least three plaintext pairs to be filtered because there are $2^{7.57}$ of difference traces.

We can classify the differential pattern of each filtered plaintext pair by analyzing their difference trace. For example, if the first column is inactive in the third round SB operation, the plaintext pair will have a differential pattern in which the s_0 of the first round output is inactive. Figure 2 shows the difference between the power traces of the plaintext pair following Figure 1. It is easy to see that $s_0 \sim s_3$ cells are inactive and $s_4 \sim s_{15}$ cells are active.

Key-recovery Each differential pattern has $2^8 - 1$ differential trails capable of pessimal diffusion. Among these differential trails, we exclude trails that do not occur through a valid differential transition from the plaintext pair. Since there are 32,385 differential transitions of the AES S-box, it is valid with a probability of $32385/2^{16} \approx 2^{-1}$; thus, the number of differential trails can be reduced to approximately $(2^8 - 1) \times 2^{-1-1} \approx 2^6$ according to the difference between plaintext pairs. For each valid differential trails, we can determine two

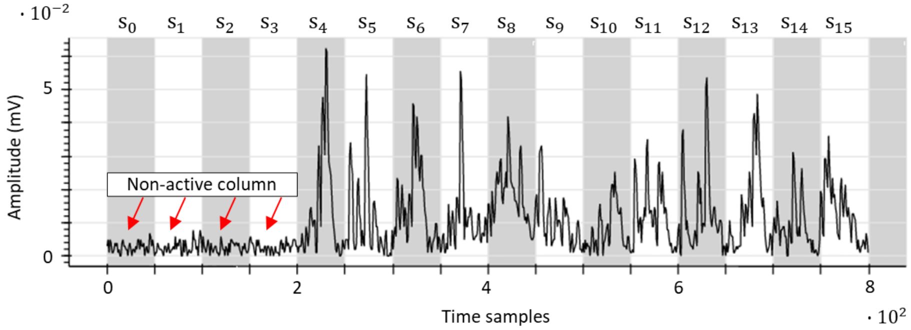

Fig. 2. Difference trace at the third round *SB* operation (The highlighted parts show that the first column is inactive).

or four 1-byte key candidates. Consequently, we obtain at most $2^6 \times 4 \times 4 = 2^{10}$ 2-byte key candidates.

By repeating this process for three plaintext pairs, the expected number of 2-byte key candidates is $2^{16-6-6-6} = 2^{-2}$. Therefore, we can recover the right 2-byte key of the master key.

Attack complexity For AES-128, we can recover the entire master key by performing the same attack on each of the eight types. Accordingly, $2^{4.32}$ encryptions and side-channel observations at the third round are needed to perform an attack on a single type. Thus, the attack requires $8 \times 2^{4.32} = 2^{7.32}$ chosen plaintexts and a time complexity of $2^{7.32}$. We need 2^{11} bytes of memory space because it stores up to 2^{10} of 2-byte key candidates.

This attack can easily be applied to AES-192 and AES-256. After the recovery of the first round key, we can now apply the types beginning from the second round and observe the difference traces at the fourth round. The second round key can be recovered by repeating the same attack. Thus, the attack requires $16 \times 2^{4.32} = 2^{8.32}$ chosen plaintexts, a time complexity of $2^{8.32}$, and 2^{11} bytes of memory space.

We tested this attack 10,000 times and found that we can collect 3 pairs of plaintext pairs when using 15 ($2^{3.9}$) plaintexts on average. From the three pairs of plaintexts belonging to a type, we could reduce the number of 2-byte master key candidates to an average 1.08.

4 SITM Attack on AES-256 Using Related-Key Differential Trail

This section presents a method of recovering a 2-byte master key of AES-256 using side-channel observation and related-key differential trail.

4.1 The Related-key Differential Trail

The related-key differential trail of AES we use is shown in Figure 3, which is a part of the multicollision trail proposed in [5]. In this related-key differential trail, there is difference only in the master key, not in the plaintext. Therefore, we search for a plaintext that satisfies the related-key differential trail existing with a probability of 2^{-30}. We can determine whether or not the related-key differential trail is satisfied by observing the difference trace of the seventh round SB operation.

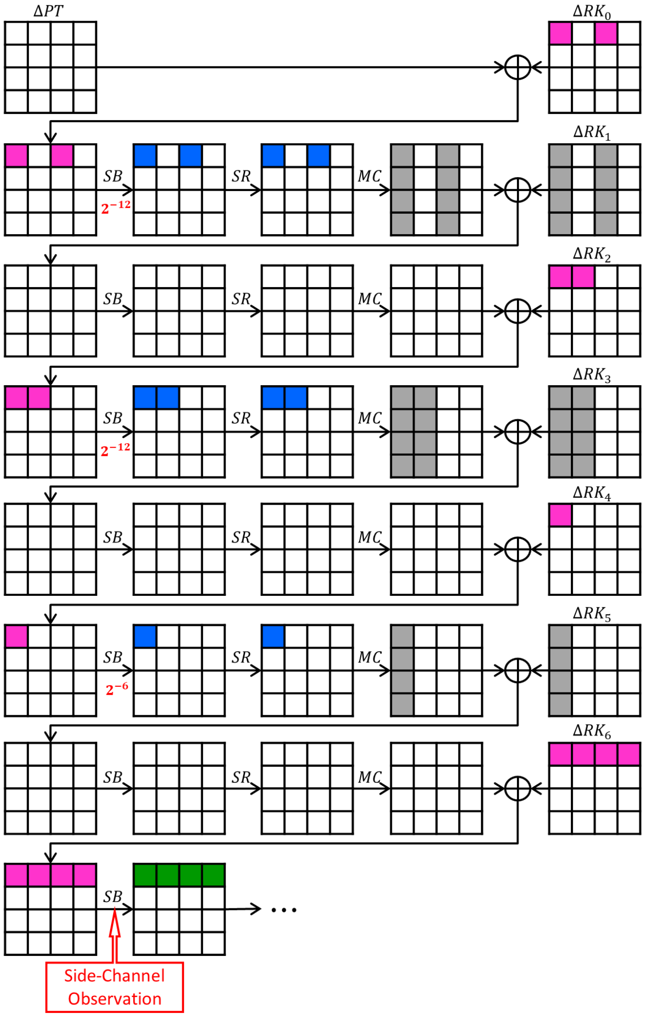

Fig. 3. The related-key differential trail of AES-256 with probability 2^{-30}. (The same colored cells have the same difference, except for the green and grey. Gray columns are the diffusion results of blue cells. Green denotes arbitrary difference.)

4.2 Application of SITM

Our SITM attack process is divided into two processes: 1) finding a plaintext satisfying the related-key differential trail; and 2) key-recovery.

Finding a plaintext satisfying the related-key differential trail

1. Generate a random plaintext.
2. Encrypt the plaintext using the master key and collect the power traces of the seventh round SB operation.
3. Encrypt the plaintext using the master key with difference and collect the power traces of the seventh round SB operation.
4. Check whether or not the difference trace is active only in the first row (Figure 4). Collect the plaintext if it is.
5. Repeat all steps until two plaintexts are collected.

We expect to collect two plaintexts by repeating the process for 2^{31} times.

Figure 4 shows the difference between the power traces of two encryptions satisfying the related-key differential trail. It is easy to see that only s_0, s_4, s_8, and s_{12} cells are active at the seventh round SB operation.

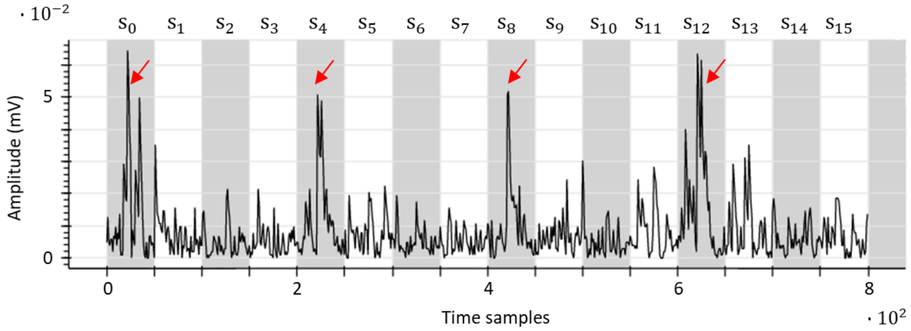

Fig. 4. Difference trace at the seventh round SB operation (The highlighted parts show that the first row is active).

Key-recovery The actual value of the s_0 and s_8 cells of the first round SB input has four candidate, respectively. We can XOR the s_0 and s_8 cells of the collected plaintext with those candidates, and obtain 2^4 2-byte key candidates. We have collected two plaintexts; hence, we can independently obtain 2^4 2-byte key candidates twice. The expected number of 2-byte key candidates is $2^{16-12-12} = 2^{-8}$, therefore, we can recover the 2-byte key of the master key.

Attack complexity The attack requires 2^{31} chosen plaintexts, 2^{32} times of encryptions and side-channel observations at the seventh round, and 2^5 bytes of memory space.

We tested this attack 100 times and found that we need an average of $2^{30.96}$ plaintexts to collect two plaintexts. From 2 plaintexts satisfying the related-key differential trail, we could reduce the number of 2-byte master key candidates to an average 1.

5 Conclusion

Our study shows that the SITM attack with the third round side-channel observation proposed in [3] can be improved. Our attack reduced the data and time complexities compared to the previous work.

We have shown that the SITM attack is possible in the related-key model and can be conducted with practical complexity. Shivam Bhasin et al. recommended a 12-round masking for AES-256 to mitigate SITM [3]. However, AES-256 requires full round masking to mitigate the SITM attacks in the related-key model because our attack uses power traces from the seventh round.

References

1. Chipwhisperer-lite xmega. https://www.newae.com/products/NAE-CW1173, accessed: 2021-08-14
2. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side-channel(s). In: CHES 2002. LNCS, vol. 2523, pp. 29–45. Springer (2002)
3. Bhasin, S., Breier, J., Hou, X., Jap, D., Poussier, R., Sim, S.M.: SITM: see-in-the-middle side-channel assisted middle round differential cryptanalysis on SPN block ciphers. CHES 2020 pp. 95–122 (2020)
4. Biham, E., Shamir, A.: Differential cryptanalysis of the data encryption standard. Springer Science & Business Media (2012)
5. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and related-key attack on the full AES-256. In: CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer (2009)
6. Breier, J., Jap, D., Bhasin, S.: SCADPA: side-channel assisted differential-plaintext attack on bit permutation based ciphers. In: 2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018. pp. 1129–1134. IEEE (2018)
7. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer (1996)
8. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer (1999)
9. Rijmen, V., Daemen, J.: Nist fips pub. 197: Advanced encryption standard (aes). Federal Information Processing Standards Publications (2001)
10. Tillich, S., Herbst, C., Mangard, S.: Protecting AES software implementations on 32-bit processors against power analysis. In: ACNS 2007. LNCS, vol. 4521, pp. 141–157. Springer (2007)