ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 8(S5), 91-97, March 2015* DOI: 10.17485/ijst/2015/v8iS5/61626

The Effect of Eight-Shuffling AES Implementations Techniques against Side Channel Analysis

Sang-Su Baek[1], Yoo-Seung Won[2], Dong-Guk Han[3*] and Jae-Cheol Ryou[4]

[1]Mobile Security Team, Solacia Inc., Seoul, South Korea;
baek@sola-cia.com
[2]Department of Financial Information Security, Kookmin University, Seoul, South Korea;
mathwys87@kookmin.ac.kr
[3]Department of Mathematics, Kookmin University, Seoul, South Korea;
christa@kookmin.ac.kr
[4]Department of Computer Science and Engineering, Chungnam National University, Chungnam-si, South Korea;
jcryou00@gmail.com

Abstract

Even though cryptographic algorithms embedded on physical devices guarantee theoretical security, they are vulnerable to side channel attacks that analyze correlations related to physical information such as power consumption and electromagnetic waves. Physical devices without any countermeasures are vulnerable to side channel analysis. The masking and shuffling techniques the most used countermeasures against side channel analysis. Masking techniques rely on the masking order, however, these techniques have a high computational cost. Shuffling techniques, on the other hand, are able to provide security without high computational cost. Recently, instead of using one countermeasure alone, a combination of them has been employed while still affording provable security at a relatively computational cost. Computational security is related to the complexity of shuffling when a shuffling technique has been employed. In this paper, we apply shuffling techniques of the Advanced Encryption Standard (AES) in a new way. Our technique involves to eight different implementations of AES. If our technique is proven safety, then we will combine masking techniques and our technique. So, we examine the theoretical versus experimentally analyzed number of power traces for the recovery key. Theoretically, our results show 64 times more shuffling complexity than a non-shuffling AES implementation. Experimentally, however, it has seven times greater shuffling complexity.

Keywords: Countermeasure, Shuffling Technique, Side Channel Analysis

1. Introduction

The secret key of a cryptographic algorithm must never be exposed to attackers. In order to prevent a secret key leakage, a cryptographic algorithm must be analyzed. The security of a cryptographic algorithm must be validated mathematically using statistical or algebraic analyses or by side channel analysis[1], which is one type of non-cryptographic analysis more recently considered. Side channel analysis exploits extra sources of information embedded on physical devices, such a calculation time, power consumption, and electromagnetic waves, during

* *Author for correspondence*

the execution of the cryptographic algorithm. For secure implementation of cryptographic algorithms against side channel analysis, countermeasures have actively been studied by cryptographic researchers. The most widely utilized countermeasures of cryptographic algorithms against side channel analysis are masking techniques and shuffling techniques.

Masking techniques[2] make the power consumption of a cryptographic device independent of the intermediate values of a cryptographic algorithm. In addition, these techniques randomly split every intermediate value occurring in the computation into d+1 shares, where d is the masking order. On the other hand, shuffling techniques[3] are executed by breaking the link between the power consumption of the device and the processed data values. Each of these techniques provides computational security. While a higher masking order guarantees security, it does not provide low computational cost. To counteract this weakness, a combination of masking and shuffling techniques is used to provide efficiency and security against side channel analysis. In particular, the combination of first-order masking and shuffling techniques is less costly than higher-order masking techniques. It has recently been shown, however, that this combination is vulnerable to biasing power analysis[4]. As such, a practical countermeasure is necessary. In the case of the Advanced Encryption Standard (AES)[5], second-order (third-order) masking is 23 (30~50) times slower than the non-masking AES; hence, masking-only techniques are not as efficient[6–8].

In this paper, we apply shuffling techniques in a new way. If these new techniques guarantee efficiency and security, then we can use them in combination with masking techniques. But in terms of security, we demonstrate that these shuffling techniques are not secure. Rather than applying general shuffling techniques at the functional level, new shuffling techniques are applied at the algorithmic level consisting of eight different implementations of AES. The eight different implementations is broken into two main groups. One group is implemented by raw version, the other is implemented by macro version. The raw version is composed of the four implementation for AES: Original 1, Original 2, Bertoni[10], and T-Table[11]. The eight-shuffling means that one of eight implementations is operated by using previously the generated random value. This approach will be expected to operate faster than the shuffling method of functional level. However, we will first have to evaluate the shuffling of algorithmic level in terms of security.

Theoretically, we expected our new shuffling techniques to offer 64 times more complexity of shuffling. However, experimental results showed 7 times more complexity of shuffling than for non-shuffling AES implementations. This results present that our new shuffling techniques only provide the complexity of shuffling $\sqrt{7}(\approx 2.6)$. We would offer good sights for embedded system designers to implement new shuffling techniques. That is, if designers apply similar countermeasures such as our techniques, these will be more vulnerable than original shuffling techniques.

The remainder of this paper is organized as follow: Section 2 describes the correlation power analysis and the shuffling techniques of AES, Section 3 introduces new shuffling techniques and discusses experimental results, and finally, Section 4 concludes this paper.

2. Correlation Power Analysis and Shuffling Techniques of AES

2.1 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known as Rijndael, is a symmetric cipher composed of a Substitution-Permutation Network (SPN) structure. For an N-bit, SPN-type, symmetric cipher of r rounds, each round consists of four transformations: the SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations. The SubBytes transformation is a non-linear byte substitution that operates independently on each byte. In the ShiftRows transformation, the bytes in the last three rows are cyclically shifted. The MixColumns transformation operates column by column over $GF(2^8)$ .Finally, in the last transformation, AddRoundKey, a round key is added by a simple bitwise XOR operation.

2.2 Correlation Power Analysis of AES S-box

2.2.1 Power Consumption Models

Most power consumption models[9] found in the literature are based on the Hamming weight model or the Hamming distance model. The Hamming weight model is based on the fact that bit one has more power consumption than bit zero, whereas the Hamming distance model bases power consumption on the number of substitutions required to

change the output data from the input data. Simply put, the Hamming weight model is the count of bit one and the Hamming distance model is the number of switching bits.

2.2.2 Correlation Power Analysis of S-Box Output

Power analysis is a powerful attack carried out through statistical analysis. If an attacker guesses the correct key, power consumption becomes proportional to the power model. Correlation Power Analysis (CPA)[9], based on the above property, calculates the correlation coefficient between the computed data and the measured power consumption of data. The highest peak of the correlation plot gives the correct key hypothesis. In general, key hypothesis is conducted byte by byte. The guessing key of one byte is the number 256. Then, we calculate correlation coefficient that Hamming weight of intermediate value which is AES S-Box output between power traces. The highest value of the correlation coefficient is the correct key.

2.2.3 Shuffling Techniques of AES

In general, shuffling techniques operate randomly at the functional level of the AES algorithm, e.g., at the SubBytes, ShiftRows, MixColumns, and AddRoundKey transformation. Shuffling consists of spreading the signal containing sensitive information about a variable X over t different signals $S_1, \ldots, S_t$, which are then leaked at different times. If the spread is uniform, the probability that S_i responds to the manipulation of X is $\frac{1}{t}$. In order to recover the key in this case, the square of the shuffling complexity t, is theoretically the number of power traces required in order to recover the correct key. Consequently, if non-shuffling techniques of AES require m power traces for recovering the secret key, shuffling techniques of AES require $m \times n^2$ power traces.

3. Correlation Power Analysis for Eight-Shuffling AES Techniques

3.1 Eight-Shuffling AES

The eight-shuffling AES techniques are listed in Table 1.

Table 1. Eight-Shuffling AES implementations

AES implementation method	Feature
Original 1	S-Box Lookup Table 256×1
Original 2	S-Box Lookup Table 16×16
Bertoni	Suggested method by Bertoni
T-Table	The Round Transformation
Original 1 Macro	AES Original 1 Macro implementation
Original 2 Macro	AES Original 2 Macro implementation
Bertoni Macro	AES Bertoni Macro implementation
T-Table Macro	AES T-Table Macro implementation

Furthermore, the pseudo code is as follow.

Table 2. The Pseudo Code of Eight-Shuffling

Input	Byte PT[16], Byte MK[16]
Output	Byte CT[16]

```
Begin
// The assignment of Round Key
Byte RK[11][16]
// The generation of Random Value
shuffle = rand()%8
// The execution of Key Expansion
KeyExpansion(Byte Masterkey[16], Byte Roundkey[11][16])
// The assignment of the execution of eight-shuffling AES
(*ptr_shuffle[8])(Byte PT[16], Byte RK[11][16], Byte CT[16])
ptr_shuffle[0] = &AES_Original1
ptr_shuffle[1] = &AES_Original1_Macro
ptr_shuffle[2] = &AES_Original2
ptr_shuffle[3] = &AES_Original2_Macro
ptr_shuffle[4] = &AES_Bertoni
ptr_shuffle[5] = &AES_Bertoni_Macro
ptr_shuffle[6] = &AES_T-Table
ptr_shuffle[7] = &AES_T-Table_Macro
// The execution of eight-shuffling AES
ptr_shuffle[shuffle](PT, RK, CT))
End
```

3.2 Correlation Power Analysis for Eight-Shuffling AES Implementation Techniques

Our experiments were carried out using an AT mega 128 board having a dedicated 8 bit processor. For the CPA experiment, we targeted the output of S-Boxes. The AES collected 100,000 power traces of the eight-shuffling AES techniques at an oscilloscope sampling rate of 250MS/s. For analyzing the non-shuffling AES, we collected 12,500 (i.e. 100,000/8) power traces of each of the eight non-shuffling AES techniques.

Figures 1 through 5 plot the value of the points of the power versus the indicated the correlation coefficient. We focus on the 6th byte of a total of 16 bytes. Figure 1 shows the result of CPA for shuffling AES. Figure 2, on the other hand, shows the analysis of CPA for each of the eight non-shuffling AES techniques at the sixth byte.

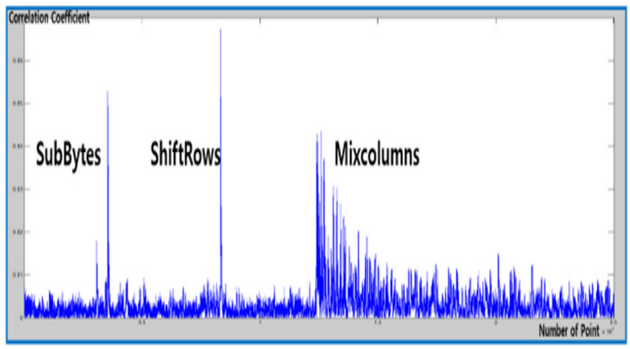

Figure 1. Results of CPA for eight-shuffling AES.

Figure 1 shows three distinct peak points; specifically, SubBytes, ShiftRows, MixColumns. To analyze these peak points, we collected each power trace. Figures 2 through 4 show the analysis of CPA for each of the eight non-shuffling AES implementations at the sixth byte.

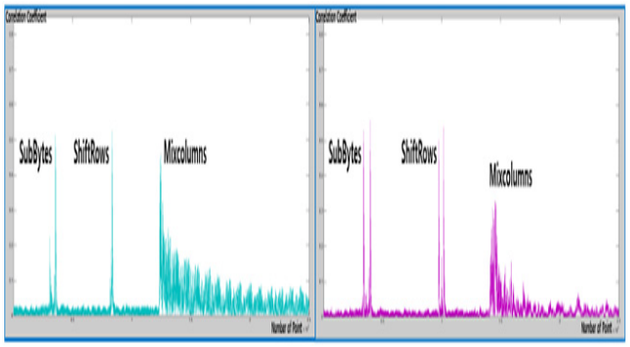

Figure 2. Results of CPA AES Original 1(Left) and AES Original 1 Macro(Right).

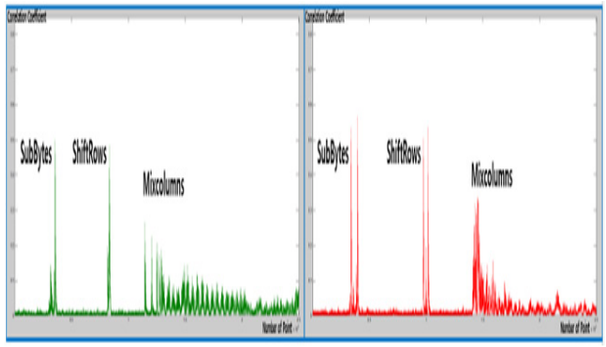

Figure 4. Results of CPA AES Bertoni(Left) and AES Bertoni Macro(Right).

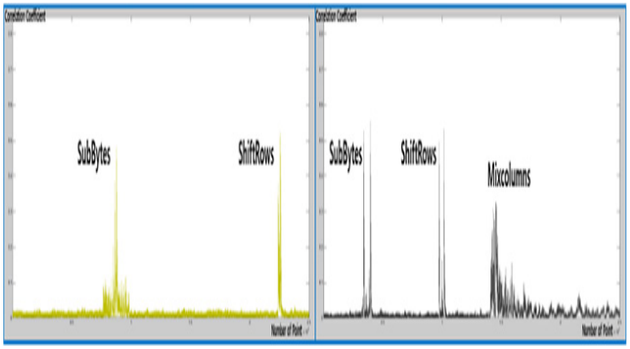

Figure 3. Results of CPA AES Original 2(Left) and AES Original 2 Macro(Right).

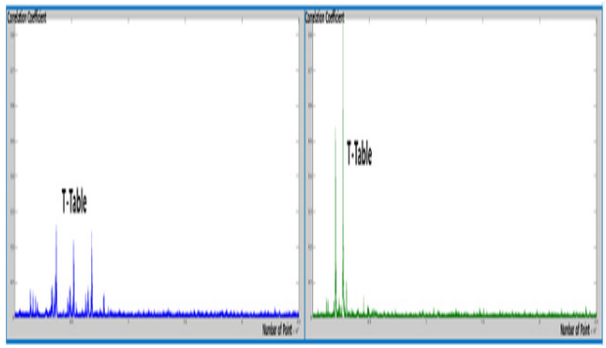

Figure 5. Results of CPA AES T-Table(Left) and AES T-Table Macro(Right).

3.2.1 Peaks of the Correlation Coefficients for SubBytes

The target position of the CPA attack is the SubBytes related to five AES implementations. The first peaks of AES Original 1, Bertoni, Original Macro, Original 2 Macro, and Bertoni Macro mean that the SubBytes transformation executed. The remainder of the AES implementations (AES Original 2, T-Table, and Original 1 Macro) leads directly to noise at the points. In spite of the different AES implementations, operational position of the intermediate value is nearly the same.

3.2.2 Peaks of the Correlation Coefficients for ShiftRows

The second peak of Figure 1 is similarly located to the second high peak, or correlation coefficient value, of AES Original 1, Original 2, and Bertoni. The highest correlation coefficient of Figure 3 is executed with S-Box output. When Figure 3(Left) is analyzed at the SubBytes, Figures 2 and 4(Left) are executed with ShiftRows. Therefore we analyzed the secret key with smaller traces than theoretically expected. Even though ShiftRows of Figures 2, 3, 4(Right) are executed at the same points, when eight-shuffling AES is analyzed, ShiftRows could not be analyzed due to it behaving as noise.

3.2.3 Peaks of the Correlation Coefficients for MixColumns

The MixColumns part is analyzed to have a form of tailbacks except for AES Original 2, T-Table, and T-Table Macro. This shape of analysis results from MixColumns being misaligned when calculating field operation. As a result, eight-shuffling AES has also shown peaks by MixColumns as seen in Figure 2 (Left, Right), 3(Right) and 4(Right).

Each power trace showed a maximum correlation coefficient value at the same points. This means that in spite of the different AES implementations, SubBytes of AES was performed at the same time. More precisely, the intermediate value of S-Box output was executed at the same moment. Because of this phenomenon, we attempted analysis with fewer power traces than theoretically needed. These results are shown in Table 3.

3.3 Performance Analysis for Eight-Shuffling AES

If the number of analyzed traces is m for non-shuffling AES and complexity of shuffling is $\frac{1}{n}$, then theoretically, $\mathbf{m \times n^2}$ traces are needed to analyze traces to which eight-shuffling AES has been applied.

Table 3 shows the minimum number of traces by which eight-shuffling AES and non-shuffling AES can be analyzed in order to determine the secret key at the sixth byte.

Table 3. The number of power traces needed for analysis

AES implementation method	The number of power traces for analysis
Original 1	200
Original 2	200
Bertoni	200
T-Table	4,300
Original 1 Macro	100
Original 2 Macro	100
Bertoni Macro	100
T-Table Macro	100
Average	662.5
The number of theoretical power traces needed for analysis	42,400
The number of experimental power traces needed for analysis	6,300

We calculate the minimum number of power traces needed for eight-shuffling AES implementation. The average of Table 3 gives 662.5 as the minimum number of power traces needed for each of the eight implementations; that is (200+200+200+4,300+100+100+100+100)/8. As the complexity of shuffling is eight in this paper, theoretically, the number of power traces for eight-shuffling AES should be 64 times more than the average minimum number of power traces needed for non-shuffling AES; namely, $662.5 \times \mathbf{8^2}$ =42,400 power traces.

It can be shown that the minimum number of power traces by which eight-shuffling AES can be analyzed is approximately 1/7 of 42,400 (i.e. 42,400×1/7≈6,300). In other words, if eight kinds of implementation methods are applied randomly, the complexity of the attack cannot be increased to the theoretical 64 times, and eight-shuffling AES can be analyzed with much less complexity of the

attack, i.e., 7 times. In other words, the complexity of shuffling is $\sqrt{7}(\approx 2.6)$.

Also, we can calculate success rate of recovering all bytes in Figure 6. That is, if we find 16 round keys, we can gain 100% success rate. When we analyze the eight-shuffling AES and non-shuffling AES, we only use 12,500 power traces. In Figure 6, light gray line represents the average success rate of non-shuffling AES. We can also calculate the success rate of eight-shuffling AES. Within 12,500 power traces, we cannot gain 100% success rate in terms of eight-shuffling AES. Therefore, in view of recovering 15 round keys, we will compare between success rate of eight-shuffling AES and non-shuffling AES. We can gain 15 round keys at around 10,000 power traces when we analyze the eight-shuffling AES. In the average of success rate of non-shuffling AES, 15 round keys are found at around 1,500 power traces.

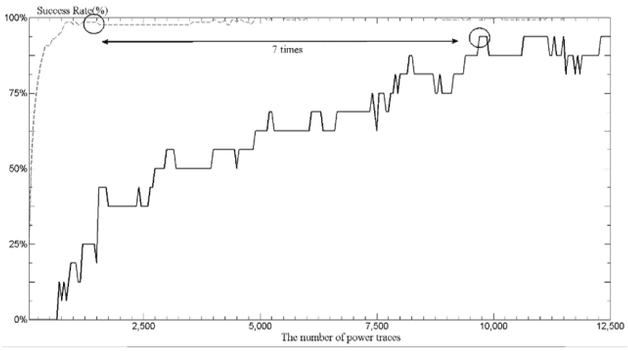

Figure 6. The comparison between success rate of non-shuffling AES and success rate of eight-shuffling AES.

The reason why eight-shuffling AES can be analyzed with fewer power traces than the theoretical number of power traces is in Table 3. In order to obtain 64 times more theoretical complexity of the attack using eight kinds of implementation methods, power information about the intermediate value of S-Box output at the analysis position must exist at different points in the eight kinds of implementations. However, as seen above in Figures 1 through 5, some of power traces of SubBytes, ShiftRows, MixColumns, which use information of the intermediate value related to S-Box output, exist at the same points. Excepting Figure 3(Left) and Figure 5(Left), the SubBytes operation of AES is occurred at the same time. So, we can divide into three groups. The first group belongs to Figure 3(Left), the second group has Figure 5(Left), and last group includes the remainders. That is, the complexity of eight-shuffling AES is three.

Consequently, the analysis could still be successful with a much less complex attack than one with theoretical complexity 64 times greater.

4. Conclusion

In this paper, we studied the effect of eight-shuffling AES implementation techniques on CPA. In the case of the eight-shuffling AES methods, the theoretical complexity of the attack was 64 ($=8^2$) times greater than that of non-shuffling AES implementations. Experimentally, however, the complexity of the attack was seven. Because the intermediate value of S-Box output was executed at the same time, we obtained a complexity of seven less than the theoretical complexity of shuffling. Therefore, with respect to side channel analysis, the algorithmic level of shuffling is no more efficient than functional level of shuffling.

5. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(NRF-2013R1A1A2A10062137)

6. References

1. Kocher P, Jaffe J, Jun B. Differential power analysis. Proceedings of CRYPTO' 99, Lect Notes Comp Sci. 1999; 1666:388–97.
2. Rivain M, Prouff E. Provably secure higher-order masking of AES. Proceedings of CHES 2010, Lect Notes Comp Sci. 2010; 6225:413–27.
3. Veyrat-Charvillon N, Medwed M, Stephanie, Standaert F. Shuffling against side-channel attacks: a comprehensive study with cautionary note. Proceedings of ASIACRYPT 2012, Lect Notes Comp Sci. 2012; 7658:740–57.
4. Cho JW, Han DG. Security analysis of the masking-shuffling based side channel attack countermeasures. Journal of SERSC: IJSIA. 2012; 6(4):207–14.
5. National Institute Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard (AES). 2001 Nov. Available from: http://www.nist.gov/manuscript-publication-search.cfm?pub_id=901427.
6. Kim HS, Hong SH, Lim JI. A fast and provably secure higher-order masking of AES S-Box. Proceedings of CHES 2011, Lect Notes Comp Sci. 2011; 6917:95–107.
7. Fumaroli G, Martinelli A, Prouff E, Rivain M. Affine masking against higher-order side channel analysis.

Proceedings of SAC 2010, Lect Notes Comp Sci. 2010; 6544:262–80.
8. Goubin L, Martinelli A. Protecting AES with shamir's secret sharing scheme. Proceedings of CHES 2011, Lect Notes Comp Sci. 2011; 6917:79–94.
9. Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model. Proceedings of CHES 2004, Lect Notes Comp Sci. 2004; 3156:16–29.
10. Bertoni G, Breveglieri L, Fragneto P, Macchetti M, Marchesin S. Efficient software implementation of AES on 32-Bit Platforms. Proceedings of CHES 2002, Lect Notes Comp Sci. 2002; 2523:159–71.
11. Fischer V, Drutarovsky M. Two methods of rijndael implementation in reconfigurable hardware. Proceedings of CHES 2001, Lect Notes Comp Sci. 2001; 2162:77–92.
12. Rivain M, Dottax E, Prouff E. Block ciphers implementations provably secure against second order side channel analysis. Proceedings of FSE 2008, Lect Notes Comp Sci. 2008; 5086:127–43.
13. Tunstall M, Whitnall C, Oswald E. Masking tables-an underestimated security risk. Proceedings of FSE; 2013.
14. Ishai Y, Sahai A, Wagner D. Private circuits: securing hardware against probing attacks. Proceedings of CRYPTO 2003, Lect Notes Comp Sci. 2003; 2729:463–81.
15. Messerges T-S, Dabbish E-A, Sloan R-H. Examining smartcard security under the threat of power analysis attack. Proceedings of IEEE Transactions on Computers, Lect Notes Comp Sci. 2002; 51:541–52.
16. Waddle J, Wagner D. Toward efficient second-order power analysis. Proceedings of CHES 2004, Lect Notes Comp Sci. 2004; 3156:1–15.
17. K. Schramm and C. Paar, Higher order masking of the AES. Proceedings of CT-RSA 2006, Lect Notes Comp Sci. 2006; 3860:239–54.