An improved LEA block encryption algorithm to prevent side-channel attack in the IoT system

Jaehak Choi * Youngseop Kim †
* Dankook University, Cheonan, Korea
E-mail: wwoo0820@gmail.com Tel: +82-10-80091132
† Dankook University, Cheonan, Korea
E-mail: wangcho@dankook.ac.kr Tel: +82-10-72728060

Abstract— **Devices of IoT (Internet of Things) are limited in resources such as CPU, memory etc. The LEA (Lightweight Encryption Algorithm) was standardized as the encryption algorithm suitable for IoT devices in Korea in 2013. However, LEA is vulnerable to the side-channel analysis attack using consumed electric power. To supplement this vulnerability, masking technique is mainly used. However, in case of masking process, the implementation time is increased, losing the characteristics of speedup and lightening. This paper proposes a new and faster LEA algorithm as a countermeasure to the side-channel attack. The proposed algorithm is about 17 times faster than existing algorithms with the masking process to prevent differential side-channel attack.**

I. INTRODUCTION

Devices of IoT are limited in use of existing encryption algorithm due to restricted resources in CPU capacities, memory size, power consumption etc. Accordingly, lightweight encryption algorithms suitable for IoT devices are being researched and developed in many parts of the world [1]. In Korea, a lightweight algorithm called LEA (Lightweight Encryption Algorithm)[2] was standardized in 2013. Contrary to AES (Advanced Encryption Standard) as the international standard block cipher algorithm [3], LEA does not use a look-up table and apply arithmetic operation in bit units, so LEA is a faster, lightweight encryption algorithm. However, devices applying encryption algorithms produce not only the encrypted data but also side-channel information such as additional information on time of use, consumed power etc, inevitably. Devices applying encryption algorithms are mainly hacked by side-channel attacks [4]. Among the preventive measures for these side-channel attacks, the masking process is used most frequently, but it poses overload problems such as dramatic slowdown, etc. This paper proposes a faster encryption algorithm without the masking process as a preventive measure for side-channel attacks.

In section II of this paper, LEA as the Korean block encryption algorithm standard is explained. In section III, the side-channel attack, which is the main attack method of block encryption algorithms, is described. In section IV, the masking technique among preventive measures for side-channel attacks is shown. And in section V, a faster and lightweight algorithm preventing side-channel attacks is suggested. In section VI, the performance of existing LEA, masking processed LEA and the proposed algorithm in this paper are compared and the conclusion is stated.

II. LIGHTENING BLOCK ENCRYPTION ALGORITHM

LEA (Lightweight Encryption Algorithm)

The LEA was the block encryption algorithm developed in 2013 to provide confidentiality in a lightweight environment where high-speed processing is required such as in mobile devices, etc. The LEA performance and operation mode was established as the domestic TTA (Telecommunications Technology Association) standard. LEA is the algorithm that encrypts 128 bits of plain text, and LEA-128, LEA-192, LEA-256 modes are determined according to the length of the secret key. According to each mode, LEA is composed of 24, 28 and 32 rounds. As LEA was made to apply to lightweight environments, ARX (Addition, Rotation, XOR) arithmetic operation is processed in 32 bit unit. The ARX arithmetic operations designed for high speed activity in 32 bit platform are composed of Addition, Rotation, XOR arithmetic operation. As these arithmetic operations process in bit units without S-BOX as look-up table contrary to AES, these are arithmetic operations of the encryption algorithm optimized to restricted environment [5].

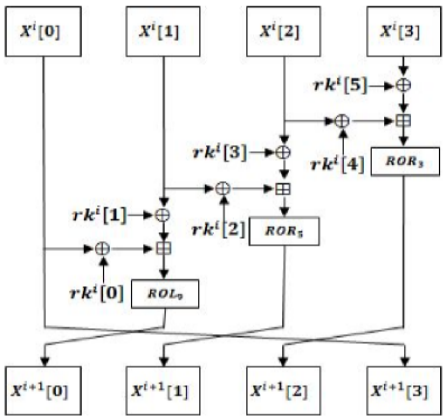

Fig. 1 LEA encrypted round function

Figure 1 represents the encrypted round function of LEA. As shown in Figure 1, this algorithm is composed of the addition of unit bit, rotation of unit bit and XOR arithmetic

operations only. In this case, the data type to be processed is processed by 4 words (32bit) divided in 4 from 1x16 byte of plain texts. The round key of LEA has the length of 192 bit regardless of mode, to be applied to round function divided in six pieces in word unit.

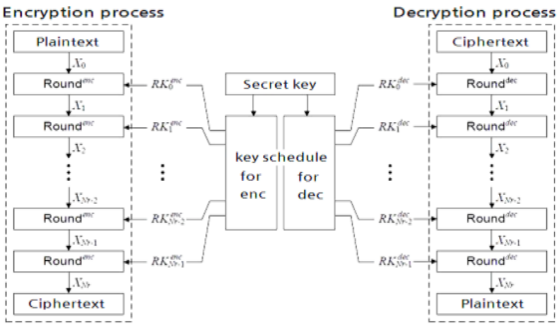

Fig. 2 Overall flowchart of LEA

Figure 2 shows the overall structure of LEA. When secret key and plain text is entered, the secret key returns mutually different 192 bit round keys at each round according to the encrypted key schedule function, and proceeds with the relevant round per mode. From Figure 2, the function implemented at each round is applied without change, with only a difference in input round key value that returned at the previous round X_{Nr-1} [6].

III. METHOD OF BLOCK ENCRYPTION ALGORITHM ATTACK

A. *Simple Power Analysis*

This is the analysis method using small number of wave patterns as one of the side-channel analyses. It is practically impossible to find out the secret key, with pattern analysis on a small number of waves. However, this method reduces the scope of additional analysis. Therefore, it is difficult to obtain the secret key directly by simple power analysis. But the scope of analysis by other hacking methods is reduced, and the time required to obtain a secret key is reduced [7].

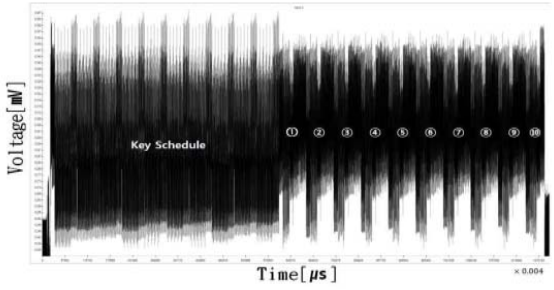

Fig. 3 Wave pattern of consumed power of encrypted device

The wave pattern in Figure 3 is the result of a simple power analysis of the AES (Advanced Encryption Standard) as the international standard for block cipher. By simple power analysis on the wave pattern above, AES algorithm is composed of 10 rounds in total, and the last round is found as different from previously performed rounds.

B. *Differential Power Analysis*

This is the mainly used side-channel analysis method, as the statistical approach analyzing multiple wave patterns. This is mainly used for the analysis of a secret key of symmetrical key algorithm. Generally, this method finds out keys partially in 8 bit units, and then finds out the overall keys comprehensively.

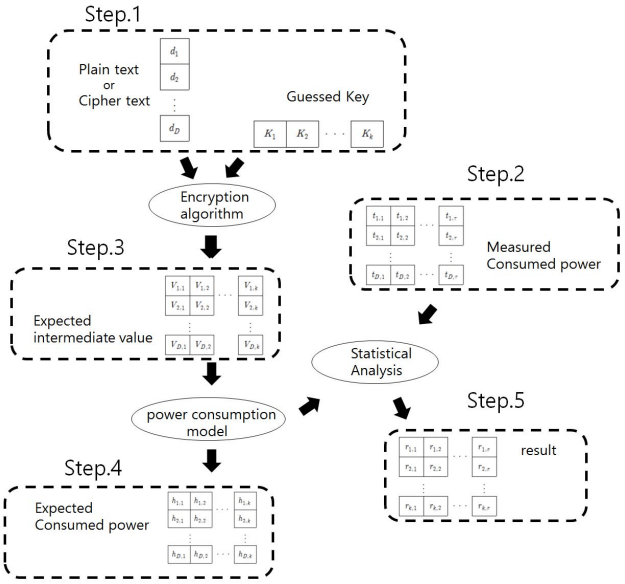

Fig. 4 Differential power analysis flow chart

Figure 4 shows the overall structure of differential power analysis. As shown in Figure 4, differential power analysis is composed of 5 stages. Firstly, attack point of the algorithm has to be established. The part composed of secret key(k) to be analyzed and information(d) manipulatable by attacker, is called an attack point. The data value composing such part is called an intermediate value (f (k,d)). The second stage collects power wave pattern. In this case, arbitrary value d as the value that can be manipulated by the attacker has to be applied to the wave pattern so collected. In order to get N wave patterns, the algorithm to be analyzed has to be proceeded with fixed k value and other mutually different N d values. The third stage is the estimation of intermediate value. In this case, the number of all cases has to be obtained. If partial key K_i of 8bit are estimated, partial key K_i can be composed of 256 kinds of cases in total from 0_x00 to 0_xff. Then, the number of estimable intermediate values is N*256. The fourth stage has to perform conversion to the power consumption model. In this case, the conversion method applicable differs according to the user environment, whether hardware as encrypted device or software transplanted to program. In case of a hardware, hamming distance [8] is applied, and in case of software, hamming weight [9][10] is applied. At the last stage, the power wave patterns obtained using the estimated intermediate value converted to the power consumption model as the result of previous stages are compared to the fixed k value. In this case, statistical method is used. The statistical method used in this case is Pearson

Correlation Coefficient. The secret key k value estimated from the power consumption model value that is the largest among N*256 power consumption model values and correlation coefficients of N power wave patterns, becomes the identical value to the secret key [11][12].

IV. PREVENTION METHOD BY SIDE-CHANNEL ATTACK

Masking technique

The differential power analysis method sets intermediate value using the arithmetic operation used in encryption algorithm, and hacks the secret key value through such intermediate value. In this case, if the intermediate value expected by an attacker differs from the intermediate value occurring from an actually moving device, then it is impossible to analyze the secret key value with differential power analysis method. In other words, this is the method to make estimable intermediate value into arbitrary value. This method arithmetically calculates the mask value arbitrarily created at every round to intermediate value.

$$v_m = v \perp m \quad (1)$$

The v is the intermediate value, m is the masking value, v_m is the masking processed intermediate value by (1). In this case, as $\perp$ is composed, the masking technique is altered. The most frequently used method is the Boolean masking technique using $\oplus$. However, masking technique has the demerit of 2 ~ 100 multiple overload incurred according to the method applied. Especially, in case of masking process on LEA, speedup process as the merit of LEA is impossible [13].

V. PROPOSED ENCRYPTION ALGORITHM

The encryption algorithm proposed in this paper is the encryption algorithm for preventing a side-channel attack, by changing data format and outputting differently from the intermediate value expected by an attacker. Since the plain text value is processed as three of each 4 bytes, we have a dummy operation of 4 bytes, in order that changed data values are assigned to each words. The proposed algorithm does not add arbitrary value during arithmetic operation like in the masking technique, but adds arbitrary value to the plain text data value. As arbitrary value is not added in arithmetic operation, but processing is implemented once only, the characteristics of existing speedup can be maintained. The data format of the proposed algorithm receives input data by 12 bytes, and encryption is implemented by 16 bytes. The 16 bytes are composed of a real data of 12 bytes and a dummy data of 4 bytes. Among 12 data bytes containing real information, 4 data bytes are selected at random and the sequence is changed mutually. And then, the changed information is stored at a dummy data of 4 bytes for subsequent deciphering. So, after applying the LEA decoder function in the decoding process, it is to restore the original plain text value by just one operation to refer the dummy data of 4 bytes. Therefore the decoder processing requires a shorter processing time than the masking technique.

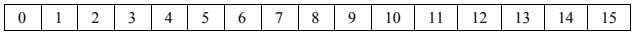

Fig. 5 LEA basic data format (1x16Matrix)

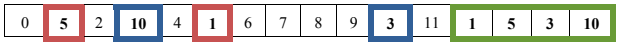

Fig. 6 Proposed data format (1x16Matrix)

Figure 5 is the data format of basic LEA algorithm. Figure 6 is the data format proposed in this paper. Figure 6 is the example of selecting 4 index values randomly and then, changing the selected index 1, 5 and index 3, 10 to store at the last 4 byte. Described in detail, the stored data of dummy byte is the index value of changed real data. The first data and second data in the dummy data are applied to index value of real data, and to store real data switched each other. The rest of the data is repeated in the same method. According to the above method, even if the input value arbitrarily by attacker is identical, power consumption wave pattern outputs differently each time. So the secret key value cannot be found out by side-channel attack. Therefore the encryption algorithm proposed in this paper is lighter and faster than the existing algorithm processed masking technique.

VI. EXPERIMENT AND CONCLUSION

LEA, masking processed LEA and the algorithm proposed in this paper, are experimented in identical environment. Each algorithm was realized by C language at Visual studio 2013, and the level of speedup was measured by measuring the time required for processing identical input value of 1,302,440 bytes.

	Processing velocity average
Masking_LEA	3.2002
Proposed LEA	0.1983

Fig. 7 Comparison of encryption process time for 1,302,440 bytes

The comparison of the processing time of the masking processed LEA and the proposed algorithm revealed about 17 degrees of difference. In addition, with the standard LEA, the encryption processing time differs by only 0.01s. Processing velocity is similar to the standard LEA as the faster encryption algorithm, and security is regarded as similar to the LEA algorithm after the masking process prevented a side-channel attack.

The algorithm proposed in this paper shows that the processing time is quicker by about 17 times than the existing algorithm for preventing a side-channel attack. Furthermore, by outputting differently from the intermediate value expected by an attacker, security is higher than the standard LEA, from differential power analysis as side-channel attack. According to the result of this experiment, the proposed algorithm is

more the speed up and lightweight algorithm for IoT device with restricted environment than the masking LEA algorithm. So our algorithm is superior in safety to the existing standard LEA.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science ICT and future Planning(2013R1A2A2A03068794).

This work was supported by the ICT Standardization R&D program of MSIP/IITP. [R0166-16-1040].

REFERENCES

1. JKIICE, *"An Efficient Hardware Implementation of Lightweight Block Cipher LEA-128/192/256 for IoT Security Applications",* Mi-Ji Sung · Kyung-Wook Shin, vol.19, No.7, Jul. 2015.
2. Korea Telecommunication Technology Association, *"128 beat light weight block cipher LEA"*, information telecommunication organization standard (Korean standard), 2013.
3. NIST, *"Advanced Encryption Standard"*, FIPS-197, Nov. 2001.
4. P. Kocher, J. Jaffe, and B. Jun, *"Differential Power Analysis."*, CRYPTO, LNCS, vol. 1666, 1999
5. PARK Je Hong, *"128beat block cipher LEA"*, TTA Journal Vol.157, 2015.
6. The Journal of Korean Institute of Communications and Information Sciences, *"Design and Implementation of Lightweight Encryption Algorithm on OpenSSL"*, Gi-tae Park, Hyo-joon Han, Jae-hwoon Lee, Vol.39B No.12, Nov.2014.
7. P. Kocher, J. Jaffe, and B. Jun, *"Introduction to differential power analysis and related attacks."*, 1998, White Paper, Cryptography Research
8. Clavier, J.-S. Coron, and N. Dabbous, *"Differential Power Analysis in the Presence of Hardware countermeasures."*, CHES 2000, LNCS. vol. 1965, pp. 252-263, 2000.
9. P. Kocher, J. Jaffe, and B. Jun, *"Differential Power Analysis."*, CRYPTO 1999, LNCS, vol. 1666, pp.388-397.
10. F.-X. Standaert, T. Malkin, and M. Yung. *"A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks."*, E UROCRYPT 2009, LNCS. vol. 5479, pp 443-461, 2009.
11. PARK Jin Hak, *"Study on side-channel analysis on light weighted block cipher LEA and design of safe responding technology"*, Kookmin University, RISS, 2015.
12. WON Yoo Seung, *"Research on safety analysis of technologies on block encryption algorithm responding to the most recent side-channel analysis"*, Kookmin University, RISS, 2013
13. Park Myung-Seo, *"Study on the Security Analysis of the Block Cipher LEA Using Side-Channel Information"*, Kookmin University, RISS, 2014
14. PARK Jin Hak, KIM Tae Jong, *"The side-channel analysis and responding method on LEA"*, Kookmin University, Journal of The Korea Institute of Information Security & Cryptology, VOL.25, NO.2, Apr. 2015.