TELKOMNIKA, Vol.17, No.3, June 2019, pp.1282~1289
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i3.9384

Power analysis attack against encryption devices: a comprehensive analysis of AES, DES, and BC3

Septafiansyah Dwi Putra*[1], Mario Yudhiprawira[2], Sarwono Sutikno[3], Yusuf Kurniawan[4], Adang Suwandi Ahmad[5]
[1]Management of Informatics, Politeknik Negeri Lampung, Bandar Lampung, Lampung Indonesia
[2,3,4,5]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia
*Corresponding author, e-mail: septa@polinela.ac.id[1], 18213024@std.stei.itb.ac.id[2]

Abstract

Cryptography is a science of creating a secret message and it is constantly developed. The development consists of attacking and defending the cryptography itself. Power analysis is one of many Side-Channel Analysis (SCA) attack techniques. Power analysis is an attacking technique that uses the information of a cryptographic hardware's power consumption. Power analysis is carried on by utilizing side-channel information to a vulnerability in a cryptographic algorithm. Power analysis also uses a mathematical model to recover the secret key of the cryptographic device. This research uses design research methodology as a research framework started from research clarification to descriptive study. In this research, power analysis attack is implemented to three symmetrical cryptographic algorithms: DES (Data Encryption Standard), AES (Advanced Encryption Standard), and BC3 (Block Cipher 3). The attack has successfully recovered 100% of AES secret key by using 500 traces and 75% DES secret key by using 320 traces. The research concludes that the power analysis attack using Pearson Correlation Coefficient (PCC) method produces more optimal result compared to a difference of means method.

Keywords*: AES, BC3, cryptography, DES, power analysis attack*

Copyright © 2019 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

The digital communication system has become one of the essential need in today's life. The rapid growth of communication technology enables the data and information to be exchanged electronically. One important factor that guarantees the exchange of information and data in a digital communication system is system security. There are a lot of methods to protect the safety of information today. The purpose of those methods is to preserve the data from the risk of tapping and theft when the exchange is taking place. Security is not an additional feature in information exchange, but a primary process of complete protection from every attacking aspect. One of many forms of guaranteeing system security is the implementation of the cryptographic algorithm. The application of a cryptographic algorithm is needed when a communication system must communicate via a media that cannot be trusted. The application of a cryptographic algorithm guarantees the confidentiality, integrity, authentication, and non-repudiation aspect of data and information [1].

Cryptanalyst, in general, assumes that a cryptographic algorithm is a purely mathematical object [2]. The assumption is based by: 1) the attacker can choose the pair of ciphertext and plaintext randomly; 2) the algorithm structure is publicly available; 3) the key is not known to the attacker. But when a cryptographic algorithm is implemented on hardware, the device is more vulnerable and more accessible to access by using physical attack techniques. Those vulnerabilities often be unnoticed by IC cryptanalysts and IC designers. Therefore, the underlying assumption from classic cryptanalyst is not possible to be used in this case. When a cryptographic algorithm is implemented on hardware, the leakage of the hardware's characteristics occurs. Some of those characteristics are time [3], sound [4], electromagnetic waves [5], and power consumption [6]. Those characteristics can be used by an attacker to recover the key used by the hardware. That leakage can't be avoided and is easily measured by adversaries by using some tools such as probe and high-frequency oscilloscope.

Received March 23, 2018; Revised January 29, 2019; Accepted February 28, 2019

The main problem that is emphasized in this research is to analyze the performance of modern encryption algorithms, those are AES, DES, and BC3, against DPA/CPA. After obtaining the measurements and results, the encryption algorithms, then, will be reviewed in its persistence from DPA/CPA attacks. The main novel aspect for this research is the performance analysis of AES, DES, and BC3 against DPA/CPA attacks. Aside from that, this research also tried to obtain the value of the secret key used by using the smallest number of traces. This publication will review the related works in the second section, the proposed method used in this research in the third section, the results and analysis of study in the fourth section and concluding remarks in the last section.

2. Related Work

Generally, attacks on cryptographic algorithm concentrate on its vulnerabilities in the mathematical aspects. IC designers assumed that if a cryptographic algorithm is secure, its implementation will also be protected. This paradigm changed when Kocher published an article about timing attack [3], and power analysis-based attack [6]. Electronic devices which implement a cryptographic algorithm with mathematical assumption could leak some precise information. The information leakage called side channel could be utilized to attain the secret key used by the cryptographic device. The implementation of a cryptographic algorithm has vulnerabilities of side channel from the physical execution of the cryptographic algorithm. The technique of bruteforcing and looking for a mathematical weakness in a mathematical algorithm is not significantly capable in attacking the secret key of a modern cryptographic algorithm [5]. The weaknesses in the form of time, power consumption, electromagnetic leakage or even the changing sound of the device could be an information source that can be exploited to slove the complexity of a cryptographic system. Some side channel techniques need technical knowledge of the internal operations of the implemented cryptographic system. Statistical methods are the most robust technique to solve the complexity of the system of cryptographic algorithm [5, 7-8].

Side channel attack (SCA) is one of attacking models that could disrupt information security when a cryptographic algorithm is implemented in a hardware [9, 10]. Differential power analysis is one of SCA techniques that could reveal secret information [11, 12]. The confidential information is the information about the secret key used in a cryptographic algorithm. The revealing process of the confidential information is procured by analyzing the leakages of information. The result of the differential power analysis (DPA) technique showed that the method could reveal 48 bits from 64 bits overall of the right secret key (75%) on DES algorithm, the rest of the key could be obtained by using the bruteforce technique. The second technique uses the correlation factor between the power traces and the hamming weight of the processed data [13]. In previous researches, the subkey of the secret key of AES and DES was obtained by using a sizeable number of power traces [13, 14]. The previous pattern of DPA attack uses roughly more than 1000 traces to get 75% of the right bits of the masterkey. The improvement from the previous attacking model was found when the correlation between the power traces and the hamming weight of the processed data is calculated and used. However, the attack utilizing the calculation of correlation must have the capability to fully control the value of the plaintext [11, 15]. Aside from the review of the attacking aspects, previous researchers had proposed several forms of countermeasure, on software or hardware level, of SCA attacks. Countermeasure technique on software level consisted of transforming and data masking [12, 16]. On the hardware level, countermeasure took the form of noise generator and desynchronizing [17-19]. Some general flaw found on the countermeasures are the performance value, and the cost deemed unfeasible from the embedded system environment [20-22].

3. Proposed Method

In this research, we proposed method power analysis attacking technique to revealing the secret key for AES, DES, and BC3 algorithm. Power analysis attacking is an attacking technique that utilizes leakage of a device's characteristics in the form of power consumption to recover the key used by the device. The leakage is measured on a sampling frequency and then be converted to a set of digital information. The measurement process produces a set of

power consumption data called power trace [12]. In this method, we carried out two attack analyses, namely DPA and CPA on encryption-based devices, AES, DES, and BC3.

3.1. Differential Power Analysis

Differential Power Analysis (DPA) is a power analysis attack that uses statistical analysis of some number of traces in its key recovery process. Secret information could not be recovered by using one or some number of power traces in DPA technique. DPA needs a lot number of power traces which is usually used in the statistical analysis that could extract even a small amount of differences between the power traces. DPA uses the formula of the difference of means (DoM) [23] as its evaluation function. The difference of means (1) is used to find the most relevant key guess from every possible guesses. *T* a set of traces and *Ti* symbolizes i-th trace so that *Ti[j]* represents power measurement at time *j* in i-th trace. *C* represents a set of known inputs or outputs. Thus *Ci* means known inputs or outputs for *i*-th trace. *b* is the value of the power consumption model. *D(Ci, b, Kn)* represents the selection function which is represented in a bit with *Ci* and *Kn,* or the key guess, as its inputs. DPA*,* generally, will produce the extreme value if the key guess K_n corresponds with the authentic key compared to the other key guesses.

$$\Delta_D[j] = \frac{\sum_{i=1}^{m} D(C_i,b,K_n)T_i[j]}{\sum_{i=1}^{m} D(C_i,b,K_n)} - \frac{\sum_{i=1}^{m}(1-D(C_i,b,K_n))T_i[j]}{\sum_{i=1}^{m} 1-D(C_i,b,K_n)} (Kocher, 1999) \quad (1)$$

3.2. Correlation Power Analysis

Correlation Power Analysis (CPA) (2) is a power analysis technique that uses Pearson Correlation Coefficient (PCC) to measure the correlation between power consumption model and the measured power traces [14, 24]. CPA was first published by Brier, et al. in 2004 on workshop Cryptographic Hardware and Embedded Systems (CHES) [13]. The value that needs to be found in CPA is the absolute value of the PCC between power consumption model and the power traces because the value could be related linearly or inversely.

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i}-\bar{h}_i).(t_{d,j}-\bar{t}_j)}{\sqrt{\sum_{d=1}^{D}(h_{d,i}-\bar{h}_i)^2.\sum_{d=1}^{D}(t_{d,j}-\bar{t}_j)^2}} \quad (2)$$

in (2), $r_{i,j}$ represents the PCC value of *i* key guess and point *j* in trace; *D* is the number of trace and *d* is the counter of the trace; $h_{i,j}$ power consumption model of key guess *i* and point *j* of trace and $\bar{h}_i$ is the average of power consumption model on key guess *i*; and $t_{d,j}$ is *d*-th trace on point *j* and $\bar{t}_j$ is the average of traces on point *j*.

4. Result and Analysis

In this research, the attacks consist of three activities; those are an attack of DES, AES and BC3 encryption devices.

4.1. Attack of DES

The attack of DES tried to guess round 16 subkeys with the assumption that if the subkey is recovered, the master key of the device could be retrieved by reverse engineering the key scheduling process and brute-forcing the rest of the key. The attack will divide the subkey into eight chunks, corresponds with the number of sboxes used in the F function in DES, so that the guess will be carried on to each six-bit subkeys of each chunk. Initially, the attack will initialize all variables needed; those variables are plaintext, ciphertext, and the power trace of the device.

Algorithm 1 represents that the attacker must find the maximum absolute value of the DoM function value for each key guess. This maximal absolute value represents the correct guess of the key among other key guesses. The correct key guess has the highest difference in the differential trace graph compared to the average of power traces.

Algorithm 1. DPA DoM for DES subkey k_{16} for each sboxes

```
Input: N pairs traces with ciphertext Ci and kg = key guess
Output: Recovered key for k16
```

```
for k_g = 0 to 255 do
        IV ← IP^-1(C_i)
        C_iR ← IV [32…63]
        for i = 0 to N do
                Matrixdata(i,k_g) ← LSB(Sbox (C_iR [numberOf Sbox] □□k_g)
                end for
        for each sample point p = 1,2,..,M of power trace do
                DoMp, k_g ← (S_δ ≡ m_i(t)|L_k0,_i = 1) - (S_δ ≡ m_i(t)|L_k0,_i = 0)
                end for
        DoMtrace, k_g ←{DoM_1, k_g, DoM_2, k_g,…, DoM_M, k_g}
        k_16[numberOfSbox] ← absmax | DoMtrace, k_g
        End
```

As shown by Table 1, the attack testing shows that the attack has succeeded to recover 48 of 64-bit of the overall key. The attack testing took 4 hours and 35 minutes to complete. The attack of DES was carried on in the 16th encryption round, specifically in the output of F function. The attack on that point enables the attacker to recover the master key used by the device by doing reverse engineering to the key schedule using the 16th round subkey and brute forcing the rest.

Table 1. DES Attack Testing Result

Variables	Results
Traces used	320
Time spent on attack execution	4 hours 35 minutes
Recovered key bits	48-bit
Unrecovered key bits	16-bit (including parity)

4.2. Attack of AES

The attack starts by initializing variables needed. The attack then loads the traces and the plaintext that will be used. The key guessing will be done in the iteration which represents the sequence of the state that will be attacked. The attack does the AES by guessing the key used in the initial AddRoundKey operation. The key guess is 8-bit sized, corresponds to a subkey that will be operated in one state. Therefore, the next iteration was programmed to guess the key with 255 (8 bit) as its limit to try every possibility of the key that fits in an AES state. Inside the iteration, the key hypothesis will be calculated by simulating the first two steps of AES, initial AddRoundKey and 1st round SubBytes.

Algorithm 2. CPA for AES master key from first round Sub Bytes

```
Input: N pairs traces with plaintext C_i and k_g = key guess
Output: Recovered key for K
for state = 1 to 16 do
        for i = 1 to number of traces do
                after_sbox[i,:] = SubBytes(bitxor(C_i[state, i], key) + 1)
                end for
        power_consumption = HW(after_sbox)
        for j = 1 to 256 do
                cmatrix[:]¬absolute(Pearson_Correlation(traces,
                power_consumption[:, j]));
                end for
        K[state] ¬ rowNumber(max(cmatrix))
        end
```

As shown in Table 2, the attack testing produced a result of 128-bit or 100% recovery for the right key. The attack testing used 500 traces and took 123.2 seconds. The master key can be directly recovered because of the vulnerability in AES which lies in the initial AddRoundKey which is basically an XOR operation between the plaintext and the masterkey.

The results of the research and test showed on Table 1 indicate that the usage of the DoM technique was considered not effective and efficient enough compared to CPA technique. It was discerned from the amount of traces used that the DoM technique needed a more significant number of traces. The computation time of DoM was longer compared to CPA. We

can generally conclude that the weak point of an encryption device is when an attacker can predict the hamming weight value of the processed data. Aside from examining the attacking aspect of CPA/DPA, this research produced some possible attack surface to be exploited. The attack surface of DPA/CPA attacking technique is the estimation of power consumption value (P_{hyp}) that has connection with the data and the power consumption. Table 3 contains the list of P_{hyp} for AES encryption algorithm on ECB (electronic code book) mode.

Table 2. AES Attack Testing Result

Variables	Results
Traces used	500
Time spent on attack execution	123,2 seconds
Recovered key bits	128-bit
Unrecovered key bits	0

Table 3. Attack Surfaces of Power Consumption on AES Encryption Algorithm

No	Attack Surface	Description
1	$P_{hyp} = HW(Sbox(P_i \oplus K_j))$	CPA *attack*
2	$P_{hyp} = HW\left(Sbox(P_i \oplus K_j)\right) mod\ 2$	DPA *attack*
3	$P_{hyp} = HW(Round9 \oplus Chipertext)$	DPA/CPA *attack from last round*
4	$P_{hyp} = HW(InvSbox(InvShifRow\ (R_{10} \oplus K_{10}) \oplus)R_{10}\ )$	CPA *attack*
5	$P_{hyp} = LSB(Sbox(P_i \oplus K_j))$	DPA *attack*

the hamming weight value was obtained by executing the following operation:

$$x = (x_1, x_2, ..., x_8) \text{ with } x_i \in \{0,1\} \quad (3)$$

$$HW(x) = \sum_{i=1}^{n} x_i \quad (4)$$

DPA/CPA attack is an attack attained from the capability of the attacker to find the power consumption (P_{hyp}).

AES encryption algorithm is a standard cryptographic algorithm capable of being combined by some other cryptographic algorithm. It is caused by its standard form so that it is possible to be adopted on a general application. AES algorithm is a symmetric cryptographic in block cipher category that can encrypt data in high speed without sacrificing security levels. However, in its implementation, AddRoundKey operation is a high vulnerability point to power analysis attack. The AddRoundKey is a final function and used to integrate key information with processed data. The input of the function is 16 bytes of state and 16 bytes of key obtained from the key expansion process. The output of this operation is the XOR bit between a state of a round of encryption and a value of the key of an expansion round of key expansion process. The XOR operation is a security gap that could be exploited in finding the secret key.

4.3. Attack of BC3

Based on the previous analysis by the researcher, the most effective way to attack a BC3 encryption device is to attack every subkey exist in every phase of BC3 [25]. If every subkey could be recovered, the attacker, after that, only must make a copy of the randomisation phase of BC3 to produce the same ciphertext by using the same plaintext used in the real BC3 encryption device. The BC3 encryption device could not be attacked by only targeting one point of its algorithm. The attack started by loading the variables needed, those are the power trace and the plaintext. After that, the attack will do an iteration to divide the key into some same-sized chunks. Because BC3 needs to be attacked in every round or every point that uses a subkey, the fragments were adjusted to the size of subkey used in round or point. Subkey used in BC3 is 32-bit in size, then if it were divided into eight chunks, the simulation will guess each four bits for every chunks or iteration.

Algorithm 3. DPA DoM for BC3 subkey KW1

```
Input: N pairs traces with plaintext Ci and kg = key guess
Output: Recovered key for k16
```

```
for group of key gk = 1 to 8 do
        for i = 1 to number of traces do
               after_sbox[i,:] = Ci[state, i]  key) + 1
               end for
        power_consumption = HW(after_sbox)
        for j = 1 to 16 do
           DoM(power_consumption, kg) ← (Sδ ≡ (ti)|Lk,i = 1) − (Sδ ≡ (ti)|Lk,i = 0)
           end for
        DoMtrace, kg ←{DoM1, kg, DoM2, kg,…, DoMM, kg}
        KW1[numberOfSbox] ← absmax | DoMtrace, kg
        End
```

To save computing resource, the attack executed by concatenating the KW1 and KW2, so the key were guessed eight bit each for all of 8 chunks. Table 4 shows that the attack testing produced a result of 16-bit recovery from 64-bit overall of KW1 and KW2 concatenated. The attack testing used 15000 traces. As shown by Table 4, the attack testing took 8300 seconds of execution.

Table 4. BC3 Attack Testing Result

Variables	Results
Traces used	15000
Time spent on attack execution	8300 seconds
Recovered key bits	16-bit
Unrecovered key bits	48-bit

One possibility of why the key recovery failed to recover all of the subkeys is the point of the attack on the algorithm. The attack point used to guess KW1 and KW2 is on the XOR operation of KW1 and KW2 to the data block. XOR is a linear operation which only changes a little amount of bit of the input to produce the output. The DPA attack, by the experience of attacking AES and DES, was succeeded to be executed by attacking a point where the function used produces a non-linear intermediate value that is the SBox function. Meanwhile, KW1 and KW2 subkeys were used in a linear function of XOR. Another possibility is the amount of traces used. The attack of KW1 and KW2 subkeys used 15000 power traces. Our literature analysis has concluded that DPA uses more trace than CPA. Let us use AES in this case; we had successfully attacked AES by using CPA with 500 traces.

On the other hand, DPA attack needed about 5000 traces minimum to attack AES [15]. CPA attack has been proved successful in attacking BC3 [14]. The attack test to BC3 was not continued after the failure to recover KW1 and KW2 subkeys. It was due to the relation between subkeys used in BC3. A round subkey in BC3 is used to generate the next round subkey. KW1 is related to the generation of K2 and K3 subkeys, while KW2 is related to the generation of K1, K3, K4, K5, and K7 subkeys.

5. Concluding Remarks

We conclude that the secret key of DES and AES encryption devices were successfully recovered. The attack of DES has successfully recovered the whole 16th round subkey. The 16th round subkey then could be used to recover the master key used by the DES encryption device. It should be concluded that the attack has successfully recovered 48 from 64-bit of a master key or 75% of the master key. The AES attack has successfully recovered 100% of the master key. The BC3 attack, on the other hand, was only able to recover 16-bit from overall 64-bit (25%) of KW1 and KW2. As shown in Figure 1, the number of traces used in an attack directly correlates to the success of the attack. The more amount of traces used in the attack resulted in higher probability of the success of the attack. While attack on DES and AES can be considered successful, the attack on BC3 has resulted in failure to obtain the secret key. The failure could be affected from two factors, those are the point of the attack that uses a linear function and the insufficient amount of power traces used. The three attacking activities proved that BC3 is the most secure algorithm to be implemented on hardware compared to DES and AES in this case. By this research, we suggest these for future works:

- The usage uses CPA rather than DPA because CPA gives more optimal results compared to DPA. Even so, other methods could be used, such as Mutual Information Analysis (MIA) or Kosmogorov-Smirnov (KS) [15].
- The attack would be better executed in a high-specification computer to get the result in the most optimal time.
- Consider implementing this attack in an asymmetric encryption algorithm or public key cryptography.

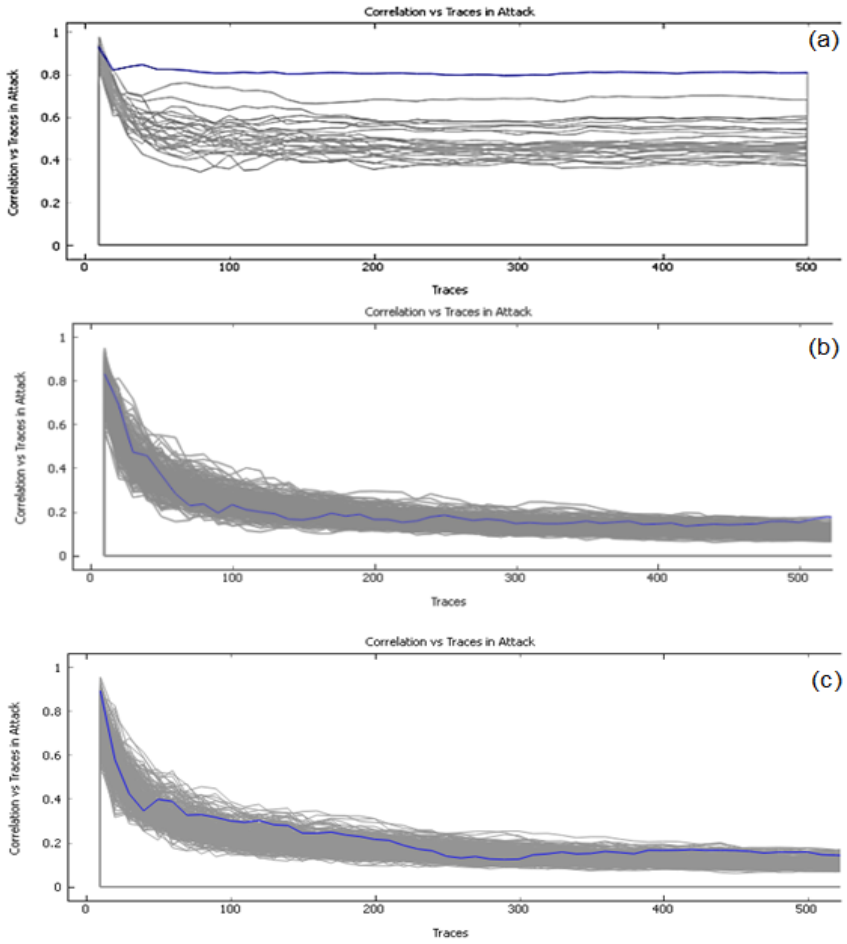

Figure 1. The number of traces used in the attack
(a) DES encryption (b) AES encryption (c) BC3 encryption

For this work, power attack was mounted using DPA and CPA technique. This will ensure the complete countermeasure against the power attacks. Another extension from this research could modify the simulation framework to analyze these types of attack and countering technique against DPA-DoM/CPA. Some of the relevant methods against DPA are the use of CAI (cognitive artificial intelligence). We will try to use a masking technique using CAI and information fusion [17, 26-29] approach.

References

[1] GC Kessler, An Overview of Cryptography. Auerbach. 1998: 65.
[2] E Hess, N Janssen, B Meyer, T Schütze. *Information leakage attacks against smart card implementations of cryptographic algorithms and countermeasures-a survey*. EUROSMART Security Conference. Marseille. 2000: 10.
[3] PC Kocher. *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems*. Annual International Cryptology Conference. Berlin. 1996: 10.

[4] GM Deepa, G SriTeja, S Venkateswarlu. An Overview of Acoustic Side-Channel Attack. *International Journal of Computer Science & Communication Networks*. 2013; 3(1): 15-20.

[5] M Masoumi, MH Rezayati. Novel approach to protect advanced encryption standard algorithm implementation against differential electromagnetic and power analysis. *IEEE Transactions on Information Forensics and Security*. 2015; 10(2): 256-265.

[6] P Kocher, J Jaffe, B Ju. *Differential Power Analysis*. Annual International Cryptology Conference. Santa Barbara. 1999.

[7] P Kocher, J Jaffe, B Jun, P Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering.* 2011; 1(1): 5–27.

[8] A Joux. Algorithmic Cryptanalisis. Boca Raton: Chapman & Hall/CRC. 2009: 158-161.

[9] GJ Orlin. The DES Algorithm Illustrated. Laissez Faire City Times. 1992; 2(28): 12-15.

[10] National Institute of Standards and Technology (NIST) Computer Security Division. FIPS 197. *Advanced Encryption Standard (AES)*. Gaithersburg: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. 2001.

[11] A Sasongko, Hidayat, Y Kurniawan, S Sutikno. Architecture for the Secret-Key BC3 Cryptography. 2011. ITB J. ICT. 5(2): 125-140.

[12] K Sakiyama, Y Sasaki, Y Li. Security of Block Ciphers: From Algorithm Design to Hardware Implementation. Singapore: John Wiley & Sons Singapore Pte. Ltd. 2015: 312.

[13] E Brier, C Clavier, F Olivier. Correlation Power Analysis with a Leakage Model. In: M. Joye and J.J. Quisquater. *Editors*. Cryptographic Hardware and Embedded Systems-CHES 2004. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004; 3156: 16–29.

[14] S Mangard, E Oswald, T Popp. Power analysis attacks: revealing the secrets of smart cards. *New York: Springer*. 2007.

[15] W Hnath, J Pettengill. Differential Power Analysis Side-Channel Attacks in Cryptography. Bachelor's Thesis. Worcester: Worcester Polytechnic Institute; 2010.

[16] E Oswald, L Mather, C Whitnall. *Choosing Distinguishers for Differential Power Analysis Attacks*. Non-Invasive Attack Testing Workshop. Nara. 2011: 14.

[17] SD Putra, AS Ahmad, S Sutikno, Y Kurniawan. Attacking AES-Masking Encryption Device with Correlation Power Analysis. *International Journal of Communication Networks and Information Security*. 2018; 10(2): 397-402.

[18] N Kamoun, L Bossuet, A Ghazel. *Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher*. 2009 3rd International Conference on Signals, Circuits and Systems (SCS). Medenine, Tunisia. 2009: 1–6.

[19] CH. Gebotys. A table masking countermeasure for low-energy secure embedded systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2006; 14(7): 740–753.

[20] Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, Cheng-Wen Wu. A high-throughput low-cost aes processor. *IEEE Communications Magazine.* 2003; 41(12): 86–91.

[21] JD Golić, C. Tymen. Multiplicative Masking and Power Analysis of AES. In: B. S. Kaliski, çetin K. Koç, C Paar. *Editors.* Cryptographic Hardware and Embedded Systems-CHES 2002. vol. 2523. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003: 198–212.

[22] N Kamoun, L Bossuet, A Ghazel. *Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher*. 2009 3rd International Conference on Signals, Circuits and Systems (SCS). Medenine, Tunisia. 2009: 1–6.

[23] S Ravi, A Raghunathan, P Kocher, S Hattangady. Security in embedded systems: Design challenges. *ACM Transactions on Embedded Computing Systems (TECS)*. 2004; 3(3): 461–491.

[24] W Gong, P Choi, BC Kim, DK Kim. *Analysis of masking effects on DPA countermeasure for lightweight cryptographic algorithms*. 2015 International SoC Design Conference (ISOCC). Gyungju, South Korea. 2015: 315–316.

[25] Ma'muri. Attack on Hardware Implementation of BC3 Encryption with Correlation Power Analysis. Master's Thesis. Bandung: Insitut Teknologi Bandung; 2018.

[26] S D Putra, AS Ahmad, S Sutikno, Y Kurniawan, ADW Sumari. Revealing. AES Encryption Device Key on 328p Microcontrollers with Differential Power Analysis. *International Journal of Electrical and Computer Engineering*. 2018; 8(6): 5144-5152.

[27] AS Ahmad, KO Bachri. *Cognitive artificial intelligence method for measuring transformer performance*. *2016 Future Technologies Conference* (FTC). San Francisco. 2016: 67–73.

[28] HRA Talompo, AS Ahmad, YS Gondokaryono, S Sutikno. *NAIDS design using ChiMIC-KGS*. 2017 International Symposium on Electronics and Smart Devices (ISESD). Yogyakarta. 2017: 346–351.

[29] CO Sereati, AD W Sumari, T Adiono, AS Ahmad. *Cognitive artificial intelligence (CAI) software based on knowledge growing system (KGS) for diagnosing heart block and arrhythmia*. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI). Langkawi. 2017: 1–5.