International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-3, Issue-2, March 2015

Boomerang Analysis of Permutation Function of 3D-AES Block Cipher

Suriyani Ariffin, Shahrizal Arshad

***Abstract*— Boomerang analysis is a type of differential attack that use two bits of key difference and comparing both to examine any differences in bit after encryption. This paper describe boomerang analysis of permutation function of three-dimensional Advanced Encryption Standard or 3D-AES. The objective of this paper is to analyze 3D-AES security using boomerang cryptanalysis. The new block cipher algorithm not been tested by boomerang analysis before. From boomerang cryptanalysis on 3D-AES block cipher, from the experiment and significance results, it can be concluded 3D-AES block cipher is secure using boomerang analysis.**

***Index Terms*— boomerang, attack, block cipher, 3D-AES, security analysis**

I. INTRODUCTION

Block cipher falls into symmetric encryption algorithms or secret-key algorithms along side with stream cipher. Concept of a block cipher algorithm is division of the plaintext into separate blocks of fixed size of bit size as in Fig. 1 such as 64 or 128 bits and encrypts each of them independently using the same key-dependent transformation [1]. There are example of successful block cipher design such as Data Encryption Standard or DES and Advanced Encryption Standard or AES which have been standardized to many different aplications and can be freely available.

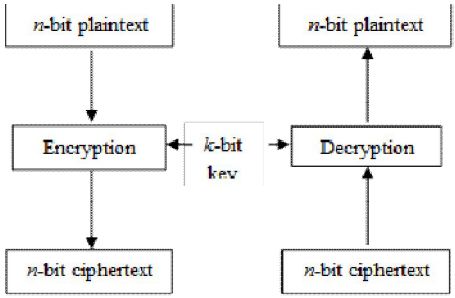

Fig. 1 : A block cipher concept

In block cipher design, it contains a sequence of simple operation which known as the round function. It is called round function because of repeated r times (r = rounds). The first round takes a n-bit of plaintext block size as input and the last round outputs the ciphertext.

Manuscript received March 25, 2015

Suriyani Ariffin, Senior Lecturer at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA.

Shahrizal Arshad, final year Master student in Computer Science at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM)

In addition, each round depends on a subkey or round key which is derived from a k-bit secret-key which known as key schedule from the derivation process. Since the recipient should be able to decode the ciphertext, the round function has to be related to any value of the secret key. To achieve the goals in decrypting ciphertext, it usually use two common way to obtain it which are Feistel ciphers and SP networks [2].

In Feistel cipher, the round function of the Feistel cipher is spliting the input block into two distinguished parts, L_{i-1} and R_{i-1} [3]. The right part R_{i-1} is unchange from it's position and it forms the left part of the output L_i. The right part of the output is created by adding a modified copy of R_{i-1} to the left part of the input L_{i-1}. It can be mathematically explain at equation below

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} + f(R_{i-1}, K_i).$$

Another approach to obtain plaintext in block cipher is using substitution-permutation networks or SP networks. In this approach, a round function is combined with layers of simple invertible functions which are substitutions or S-boxes and permutations or P-boxes. The substitution layers is acting by using small units of data which rarely more than eight consecutive bits and their nonlinear properties that high probability to introduce local confusion into the cipher. For the permutation layer, it is using simpler linear transformations, however, it operates on the complete block, thus, diffuse the effect of the substitutions [4]. The most acknowledge block cipher based on an SP network is the AES. It also note that the f-functions of many Feistel ciphers consist of a small SP network.

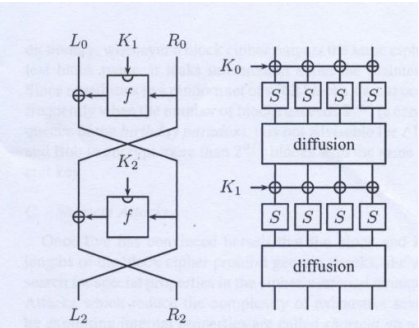

Fig. 2: Comparison of Feistel cipher (left-side) and SP network (right-side) [1].

II. BOOMERANG ANALYSIS

A. Boomerang Cryptanalysi

The boomerang cryptanalysis is based on differential cryptanalysis. Boomerang attack was discovered by David A. Wagner as an improvement in flexibility of differential cryptanalysis. The main point of boomerang attack [5] is using of two short differential characteristics which have high probabilities of bit difference on ciphertext instead of using a very long differential characteristic of low probability [6]. Assuming that a block cipher, $E: \{0,1\}^{NB} \times \{0,1\}^{NK} \rightarrow \{0,1\}^{NB}$ can be elavated as a combination of two sub-ciphers, for example, $E = E_1 \circ E_0$. NB and NK denote the block size and the key size of the cipher respectively. Assuming that a related-key differential characteristic $\alpha \rightarrow \beta$ of E_0 under a key difference ΔK_0 with probability p and another related-key differential characteristic $\gamma \rightarrow \delta$ for E_1 under key difference ΔK_1 with probability q. Thus, $p = p_k \times p_{c|k}$ where p_k is the probability that the differential characteristic way in the key schedule corresponding to E_0 that will be satisfied while $p_{c|k}$ is the probability that the differential characteristic way in the main cipher which $\alpha \rightarrow \beta$ in E_0, will be satisfied given that the key differential characteristic is satisfied as stated in [7]. Furthermore, $q = q_k \times q_{c|k}$ with same definitions with E_1. The differential characteristic trails of E_0 and E_1 are known as upper and lower trail respectively.

For boomerang cryptanalysis, it increases probability of the potential effectiveness of differential cryptanalysis. This is because it can make use of features that do not penetrate through the perfect cipher. It also has certain kinds of added complexities such as a bit transpose in the center of the block cipher, hence, it does not role as a barrier to the boomerang attack. Nevertheless, the boomerang cryptanalysis has its own restriction. It only produces a result, if both characteristics are present, thus, each characteristic restricted to do testing by independently [8]. Even though, it seems to repeat the number of rounds a cipher needs to be considered secure. In contrast, the difference between blocks need to be précised for the characteristic to be an input since at one end of a sequence of rounds. It is not possible to cascade the boomerang cryptanalysis directly to break a block cipher into four or more pieces.

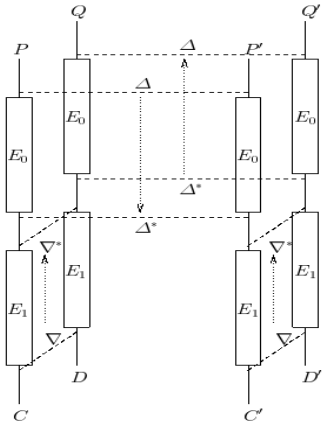

Fig. 3: A basic of boomerang cryptanalysis [5].

There are also relation due to boomerang analysis. Biryokov et. al [9] was launched the first known key-recovery attack on AES-256. From Biryokov paper, it state that it is a related-key differential attack that exploits a differential characteristic path of high probability, where both plaintext and key are allowed to have non-zero differentials. Later, Biryukov and Kovratovich [10] used a shortened version of the related-key differential characteristic to construct a distinguisher for the related-key boomerang attack on AES-256.

B. Boomerang Amplified Cryptanalysis

Another technique in boomerang cryptanalysis is boomerang amplifier cryptanalysis. Boomerang amplifier cryptanalysis operates by assuming the pairs of bit inputs, and it differentiates by using the Exclusive-OR or XOR operation that required for the characteristic of the first few rounds, as completely independent. Boomerang amplifier cryptanalysis obtaining the two pairs at same time which one can create any desired XOR operation difference between the two pairs. With this technique, boomerang cryptanalysis is allowed to be mounted with only chosen plaintext instead of the adaptive chosen ciphertext as well [11].

For comparison, the original boomerang cryptanalysis [12] requires small quantities of total queries than boomerang amplifier cryptanalysis. It is because in a boomerang amplifier cryptanalysis, the sufficient right input pairs need to be requested to determine an internal collision property that allow the desired relationship between the pairs.

In this cryptanalysis, there are three things that make the boomerang amplifier attack is useful in many attack or security testing:

1. When mounting an attack, attackers occasional need to determine key material on one end or the other of the cipher. With a chosen-plaintext/ adaptive chosen-ciphertext attack model, the number of requested plaintexts/ciphertexts increase when attackers have to guess the right key material on either end. With a chosen-plaintext only attack, key material can be guess at the end of the cipher, and not have to increase the number of chosen plaintexts requested.
2. Not on pairs only, boomerang amplifier attacks can be used on k-tuples of the text.
3. Boomerang-amplifier attack can be used to get pairs or k-tuples of texts through part of the block cipher, then covering the remaining rounds of the cipher with truncated differentials or differential-linear characteristics. Truncated differentials can be used on the specify only small part of the block and it cannot be used with a common boomerang attack [13].

II. 3D-AES BLOCK CIPHER

A. Design of 3D-AES

3D-AES block cipher is originally based from AES symmetric encryption of block cipher algorithm which is a key-alternating block cipher. It contained the rotation key function, consisting of three iterations of round function and key mixing operations. The three round functions include non-linear substitution function, permutation function and transposition function the 3D-AES block cipher block diagram [14].

The encryption process consists of rotationKey function which is the rotation key for arranging byte in 4 x 4 x 4

matrices that can be rotated in three axes contain x-axis, y-axis and z-axis design by Ariffin et. al. [15]. Second function is the 3D-SliceRotate. This function is a transformation function comprised the existing function which is SubBytes that been used in the AES symmetric encryption block cipher, getSlice and getRotateSlice. The next function is the mixColumn which is a transposition function that utilised linear transformation and also uses in AES symmetric encryption block cipher. The last function is the addRoundKey which is a key mixing function that trails every round function. Key mixing is a bitwise Exclusive-OR or XOR operation of the cipher suite with the round key. As a key iterated block cipher, there are n key mixing operation with the round key. The n keys are generated subkeys from the 16 bytes secret key using a key expansion function. It has similar function to key expansion function of the AES symmetric encryption block cipher.

For the decryption process, it consists of rotationKey function act as addition secret key for reverse the encryption process. The second function of decryption process is the 3D-invSliceRotate. 3D-invSliceRotate is a transformation function which included the exisiting invSubBytes function in AES symmetric encryption block cipher and new function called getInvSlice and getInvRotateSlice. This function is anti-clockwise direction which is inverse with the encryption process. The next function is the invMixColumns. It is a transposition function that uses a linear transformation and used in transposition function of the AES symmetric encryption block cipher too. The decryption process is to be carried out for the same number of iterations with similar subkeys in reverse order of the encryption process.

III. Design of Permutation Function

A. Array of bytes

Ariffin S. et al. [16] stated that the storage of the plaintext in the 3D-AES block cipher algorithm operation is done in three-dimensional array which is 4^3 or 4 x 4 x 4 matrices of bytes known as Cube in Fig. 6. The Cube is mapped in cube polygon with length of 64 bytes, equal to 512 bits. The permutation function contains rotation at x-axis, y-axis and z-axis which known as rotationKey. The Cube for a 64-byte data block denoted in Fig. 6 as (1) with bytes inserts column wise [16].

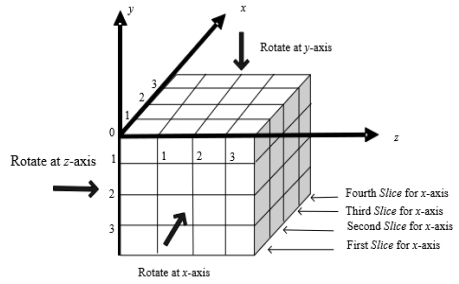

Fig. 6 : A storage format in Cube [16].

$$Cube = \left|\begin{matrix} a_{000}\,a_{001}\,a_{002}\,a_{003} \\ a_{010}\,a_{011}\,a_{012}\,a_{013} \\ a_{020}\,a_{021}\,a_{022}\,a_{023} \\ a_{030}\,a_{031}\,a_{032}\,a_{033} \end{matrix}\right| \left|\begin{matrix} a_{100}\,a_{101}\,a_{102}\,a_{103} \\ a_{110}\,a_{111}\,a_{112}\,a_{113} \\ a_{120}\,a_{121}\,a_{122}\,a_{123} \\ a_{130}\,a_{131}\,a_{132}\,a_{133} \end{matrix}\right| \left|\begin{matrix} a_{200}\,a_{201}\,a_{202}\,a_{203} \\ a_{210}\,a_{211}\,a_{212}\,a_{213} \\ a_{220}\,a_{221}\,a_{222}\,a_{223} \\ a_{230}\,a_{231}\,a_{232}\,a_{233} \end{matrix}\right| \left|\begin{matrix} a_{300}\,a_{301}\,a_{302}\,a_{303} \\ a_{310}\,a_{311}\,a_{312}\,a_{313} \\ a_{320}\,a_{321}\,a_{322}\,a_{323} \\ a_{330}\,a_{331}\,a_{332}\,a_{333} \end{matrix}\right|$$

Fig. 7 : Byte inserted in Cube [16].

The basic unit to process the 3D-AES block cipher is a byte which is refered to a sequence of eight bits treated same as a single element in the AES symmetric encryption block cipher. The three-dimensional arrays of bytes will denoted in (1) by index number:

$$byte_0, byte_1, byte_2, byte_3, byte_4, \dots byte_{63}.$$

The bytes and the bit ordering wihtin bytes are derived from the input sequence of 512 bits

$$input_0, input_1, input_2, input_3, input_4, \dots input_{511},$$

as follows:

$byte_0 =$
$input_0,$
$input_1, input_2, input_3, input_4, input_5, input_6, input_7$
$byte_1 =$
$input_8,$
$input_9, input_{10}, input_{11}, input_{12}, input_{13}, input_{14}, input_{15} byte_2 =$
$input_{16},$
$input_{17}, input_{18}, input_{19}, input_{20}, input_{21}, input_{22}, input_{23} \dots$
$byte_{63} =$
$input_{504},$
$input_{505}, input_{506}, input_{507}, input_{508}, \dots, input_{511}$

After following last input, hence:

$$byte_n = input_{8n}, input_{8n+1}, \dots, input_{8n+7} \qquad (2)$$

B. Input and Output

Ariffin S. et al. from the paper [16] also stated that the input of the encryption function includes a plaintext block, a rotation key and a secret key which their objective is to produce the output which is ciphertext block while the input of the decryption function is the ciphertext block. The cipher-state can be illustrated as a cube of bytes, with four rows and four columns. The columns number in the state is denoted by N_b , and the number of slice is denoted by N_s. From those two value, block length is equaled when it divided by 64 bytes. From [16], assuming that the plaintext block be denoted by:

$$p_0, p_1, p_2, p_3, \dots, p_{(4.N_b-1)+(16.N_s-1)}$$

where, p_0 denotes the first byte while $p_{(4.N_b-1)+(16.N_s-1)}$ denotes the last byte of the plaintext block. Assume the ciphertext block be denoted by:

$$c_0, c_1, c_2, c_3, \dots, c_{(4.N_b-1)+(16.N_s-1)}$$

where, c_0 denotes the first byte while $c_{(4.N_b-1)+(16.N_s-1)}$ denotes the last byte of the ciphertext block. Let the cipher-state be denoted by:

$$a_{i,j,k}, 0 \le i < N_s, 0 \le j < 4, 0 \le k < N_b$$

where $a_{i,j,k}$ denotes the byte in slice i, row j and column k. The input bytes are mapped onto the state bytes in order, also define from (1):

$$a_{0,0,0}, a_{0,1,0}, a_{0,2,0}, a_{0,3,0}, a_{0,0,1}, a_{0,1,1}, a_{0,2,1} \dots, a_{3,3,3}.$$

The input is plaintext block for encryption function and the mapping is:

$$a_{i,j,k} = p_{16i+j+4k}, 0 \le i < N_s, 0 \le j < 4, 0 \le k < N_b. \qquad (3)$$

The input is the plaintext block of an encryption function and the mapping is:

$$a_{i,j,k} = c_{16i+j+4k}, 0 \le i < N_s, 0 \le j < 4, 0 \le k < N_b. \qquad (4)$$

The rotation key is the extra key to rotate the cube at x-axis, y-axis and z-axis for the permutation function. The number of rotation key is denoted by N_t and the rotation key is equal to the three axes. Assuming the rotation key denoted by :

$$q_1, q_2, q_3$$

Hence :

$$Q = q_i, 0 \leq i < N_t \quad (5)$$

The cipher key obtained from the secret key is illustrated from [16] as a square array consists of four rows and four columns and mapped into one-dimensional cipher key. The columns number of cipher key is denoted by N_k and it is equal to the key length divided by 32. The bytes of the key are mapped into the bytes of the cipher key in following order :

$$k_{0,0}, k_{1,0}, k_{2,0}, k_{3,0}, k_{0,1}, k_{1,1}, k_{2,1} \dots, k_{3,3}.$$

Assuming the key denoted by:

$$s_0, s_1, s_2, s_3, \dots, s_{(4.N_b-1)}$$

Hence :

$$k_{i,j} = s_{i+4j}, 0 \leq i < 4, 0 \leq j < N_b. \quad (6)$$

C. *Permutation Function*

Ariffin et al. stated that the key-iterated block cipher utilise the same round transformation from the key-iteraing block cipher design [16]. Assume that the boolean permutation, denoted as $B[k]$ from the number of rounds by r, from k^r to k^0 to be:

$$B[k] = \sigma[k^r] \circ p^r \circ \sigma[k^{r-1}] \circ \dots \circ \sigma[k^1] \circ p^1 \circ \sigma[k^0] \quad (7)$$

where $\sigma[k^r]$ is the key addition, p^r is the r^{th} round of the round transformation of the block cipher based on the Wild Trail Strategy [17], k^r is the r^{th} round key and ° is the input to p^2 $\sigma[k^2]$ that derived from the output of p^1 $\sigma[k^1]$ and to other input gradually increasing the r value. The boolean permutation $B[k]$ is a sequence-dependent transformation of p^i $\sigma[k^i]$. Assuming the round transformation to be :

$$\rho = \theta \circ \lambda \circ \gamma$$

where γ is the non-linear function or substitution function of the slice at round function and θ is the mixColumns function from AES block cipher components and running in coumns of four bytes each. While λ is the permutation function of the slice at the round function. From (7), the boolean permutation has generated to be

$$B[k] = \sigma[k^r] \circ \theta \circ \lambda \circ \gamma \circ \sigma[k^{r-1}] \circ \dots \circ \sigma[k^1] \circ \theta \circ \lambda \circ \gamma \circ \sigma[k^0] \quad (8)$$

From (8), all round of the cipher utilise the similar round transformation. It means that 3D-AES block cipher exhibits is a key-iterated block cipher. On the other hand, a rotatiton key is used as a catalyst effect to the confusion property in the block cipher. There are also different index numbers for every slice in different axis as the rotation key from (5) regarding to the permutation function.

IV. Experimental Work

Analysis is the most crucial factor in evaluating and highlighting features of 3D-AES block cipher. The analysis from boomerang cryptanalysis is done by using bit difference of each round key pair by using MATLAB program.

A. *Bit Difference Analysis in Boomerang Cryptanalysis*

Since boomerang cryptanalysis is based from differential analysis, it attempts to generate differences between two or more plaintexts. Assume from [5] Tablethat we have two plaintexts, P, P' alongside with their respective ciphertexts Q,Q'. Let E(.) as the encryption operation and decompose the cipher into $E = E_1 \circ E_0$, where E_0, represent the first half of the cipher and E_1 represents the remaining half. From this using differential characterisitic, E_0 will generates $\Delta \rightarrow \Delta^*$ and the inverse of the differential characteristic is ${E_1}^{-1}$ for $\nabla \rightarrow \nabla^*$.

For this analysis, we using MATLAB program to determine the bit difference of each round key pair. A huge difference of bit in each round key pair can provide confusion property. It is one of the essential aspect of s of the security of encryption block ciphers which it is deals with the differences of each round key pair. It measures the differences in bit between two round key selected from a sequence of 512 bits. Experiment activitiy can be conducted using Windows operating system or open-sources operating system such as Linux. All data of each round key in sequence of 512 bits were generated and evaluated in offline mode to avoid interference from outside environment. The sequence of 512 bits were generated and tested which will be display in tables from the experiment activity done.

From the experiment analysis on the data shown in Table 5.1, Table2 and Table 5.3, the bit differences value for the first round through the tenth round for sequence of 512 bits are recorded. There are three round key pairs that show the highest difference of bits greater than standard ratio of difference to be assume secured which is 0.500. From the results of the analysis, it can be concluded that there is a relationship between the bit difference of each round key pairs and the level of security in 3D-AES symmetric encryption block cipher.

V. Result Analysis

For this paper, bit difference analysis is used to test the 512 bits of 3D-AES block cipher algorithm. The results is recorded from the experimental of bit difference testing as shown in Table 1, Table 2 and Table 3. The rate of bit differences in Table 2 and 5.3 obtained from the number divided by the total number of bits for the same round which is 512 bits.

From the Table 1, the most significant of bit difference of round keys pair are round key (3,4), (7,10), and (8,10) which are hold 312 differences in bit. The least significant of bit difference of round keys is round key (2,6) which is hold 100 difference in bit.

From the both table of Table 2 and Table 3, they are 11 pairs of round key that achieved the ratio more than 0.500. The highest rate or percentage of difference for 3D-AES are from round key (3,4), (7,10), and (8,10) which the ratio

of those three pairs are 0.6016 while the least one is from round key (2,6) which is 0.1953. The rate of difference in

Table 1 : Results for Three-dimensional Advanced Encryption Standard (3D-AES) in bits differences (*num*) from round key 1 to 10 pairs

Round Key (E_1) / Round Keys (E_0)	1	2	3	4	5	6	7	8	9	10
1		220	216	220	196	200	216	200	196	196
2	220		220	216	180	100	196	180	116	180
3	216	220		312	232	244	276	296	216	252
4	220	216	312		232	228	212	276	292	220
5	196	180	232	232		232	248	180	212	300
6	200	100	244	228	232		308	196	228	196
7	216	196	276	212	248	308		228	308	312
8	200	180	296	276	180	196	228		308	312
9	196	232	216	292	212	228	308	308		200
10	196	180	252	220	300	196	312	312	200	

Table 2 : Results for Three-dimensional Advanced Encryption Standard (3D-AES) in rate of difference (%) from round keys 1 to 5 pairs.

Round Key (E_1) / Round Key (E_0)	1	2	3	4	5
1		0.4297	0.4219	0.4297	0.3828
2	0.4297		0.4297	0.4219	0.3516
3	0.4219	0.4297		0.6094	0.4531
4	0.4297	0.4219	0.6094		0.4531
5	0.3828	0.3516	0.4531	0.4531	
6	0.3906	0.1953	0.4766	0.4453	0.4531
7	0.4219	0.3828	0.5391	0.4141	0.4844
8	0.3906	0.3516	0.5781	0.5391	0.3516
9	0.3828	0.4531	0.4219	0.5703	0.4141
10	0.3828	0.3516	0.4922	0.4297	0.5859

Table 3 : Results for Three-dimensional Advanced Encryption Standard (3D-AES) in rate of difference (%) from round keys 6 to 10 pairs.

Round Key (E_1) / Round Key (E_0)	6	7	8	9	10
1	0.3906	0.4219	0.3906	0.3828	0.3828
2	0.1953	0.3828	0.3516	0.2266	0.3516
3	0.4766	0.5391	0.5781	0.4219	0.4922
4	0.4453	0.4141	0.5391	0.5703	0.4297
5	0.4531	0.4844	0.3516	0.4141	0.5859
6		0.6015	0.3828	0.4453	0.3828
7	0.6015		0.4453	0.6016	0.6094
8	0.3828	0.4453		0.6016	0.6094
9	0.4453	0.6016	0.6016		0.3906
10	0.3828	0.6094	0.6094	0.3906	

bit obtained when the bit difference of each round key pairs is divided to 512 bit of sequence from ciphertexts.

Rate of difference (%) = $\frac{frequency\ of\ bit\ difference}{512\ bit\ from\ sequence}$

The minimum requirement for 3D-AES to assume to be secure must be the ratio is on par or greatest than 0.500 in rate of difference in bit. It can be said that the pairs of round key (3,4), (3,7), (3,8), (4,8), (4,9), (5,10), (6,7), (7,9), (7,10), (8,9), and (8,10) are secure from the boomerang cryptanalysis while the other pairs of round key are assumed to be least secure from the boomerang cryptanalysis.

The result analysis objective is tested to identify the bit independence of the 3D-AES block cipher. From the result, it is clear that the more differences in each bit of each round key pair, the more secure the round key pair that can avoid exploitation of ciphertext from boomerang attack. The permutation function of the 3D-AES block cipher has capability to conceal the ciphertext from threats. It is create confusion performance to be carried out of the non-linear relationship between the plaintext and the ciphertext.

VI. CONCLUSION

From the analysis had done, it concluded that performing boomerang cryptanalysis on 3D-AES block cipher shows significance output. It can be clearly stated that the block cipher of 3D-AES is managed to secure against conventional non related-key attacks. 3D-AES block cipher can be improved its security to resist any attack in future. The security of 3D-AES block cipher depends on how many bit difference in one sequence in differential or boomerang attack. With advancement in cryptography field, improvement of block cipher in security is important against new attacks that been developed and continuously improved in short time.

REFERENCES

[1] De Canniere, C., Biryukov, A., & Preneel, B. (2006). An introduction to block cipher cryptanalysis. *Proceedings of the IEEE*, *94*(2), 346-356.

[2] William, S. (2008). *Network security essentials*. Pearson Education India.

[3] Feistel, H. (1973). Cryptography and computer privacy. *Scientific american*, *228*, 15-23.

[4] Shannon, C. E. (1949). Communication Theory of Secrecy Systems*. *Bell system technical journal*, *28*(4), 656-715.

[5] Wagner, D. (1999, January). The boomerang attack. In *Fast Software Encryption* (pp. 156-170). Springer Berlin Heidelberg.

[6] Biham, E., Dunkelman, O., & Keller, N. (2005). Related-key boomerang and rectangle attacks. In *Advances in Cryptology–EUROCRYPT 2005* (pp. 507-525). Springer Berlin Heidelberg.

[7] Choy, J., Zhang, A., Khoo, K., Henricksen, M., & Poschmann, A. (2011). AES variants secure against related-key differential and boomerang attacks. In *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication* (pp. 191-207). Springer Berlin Heidelberg.

[8] Soleimany, H., Sharifi, A., & Aref, M. (2010, April). Improved Related-Key Boomerang Cryptanalysis of AES-256. In *Information Science and Applications (ICISA), 2010 International Conference on* (pp. 1-7). IEEE.

[9] Biryukov, A., Khovratovich, D., & Nikolić, I. (2009). Distinguisher and related-key attack on the full AES-256. In *Advances in Cryptology-CRYPTO 2009* (pp. 231-249). Springer Berlin Heidelberg.

[10] Biryukov, A., & Khovratovich, D. (2009). Related-key cryptanalysis of the full AES-192 and AES-256. In *Advances in*

Cryptology–ASIACRYPT 2009 (pp. 1-18). Springer Berlin Heidelberg.

[11] Joux, A., & Peyrin, T. (2007). Hash functions and the (amplified) boomerang attack. In *Advances in Cryptology-CRYPTO 2007* (pp. 244-263). Springer Berlin Heidelberg.

[12] Kelsey, J., Kohno, T., & Schneier, B. (2001, January). Amplified boomerang attacks against reduced-round MARS and Serpent. In *Fast Software Encryption* (pp. 75-93). Springer Berlin Heidelberg.

[13] Kim, J., Moon, D., Lee, W., Hong, S., Lee, S., & Jung, S. (2002). Amplified boomerang attack against reduced-round SHACAL. In *Advances in Cryptology—ASIACRYPT 2002* (pp. 243-253). Springer Berlin Heidelberg.

[14] Ariffin, S. (2012, October). Secure Block Cipher Inspired by The Human Immune System. School of Graduate Studies, Universiti Putra Malaysia.

[15] Ariffin, S., Mahmod, R., Jaafar, A., & Ariffin, M. R. K. (2012). Symmetric Encryption Algorithm Inspired by Randomness and Non-Linearity of Immune Systems. *International Journal of Natural Computing Research (IJNCR)*, *3*(1), 56-72.

[16] Ariffin, S., Mahmod, R., Jaafar, A., Rezal, M., & Ariffin, K. (2012). An Immune System-Inspired Byte Permutation Function to Improve Confusion Performance of Round Transformation in Symmetric Encryption Scheme. In *Computer Science and its Applications* (pp. 339-351). Springer Netherlands.

[17] Daemen, J. & Rijmen, V. (2002). AES and the Wide Trail Design Strategy. In: Knudlen, L. (ed.) Advanced in Cryptology EUROCRYPT 2002. Lecture Notes in Computer Science, vol. 2332 (pp. 108-109). Springer Berlin Heidelberg.

Suriyani Ariffin was graduated with Phd in Computer Security from Universiti Putra Malaysia (2013). She has many years working experience including in industry and in academic. In industry, she was an analyst programmer and also a consultant of various system development and maintenance projects especially in banking and government sectors. She is currently a senior lecturer at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA.

Shahrizal Arshad is currently a final year Master student in Computer Science at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM) working under Dr. Suriyani Ariffin.