SMS Authentication Code Generated by Advance Encryption Standard (AES) 256 bits Modification Algorithm and One Time Password (OTP) to Activate New Applicant Account

Eddy Prasetyo Nugroho, Rizky Rachman Judhie Putra, Iman Muhamad Ramadhan
Department of Computer Science Education, Faculty of Mathematics and Natural Sciences Education
Indonesia University of Education
Bandung, West Java, Indonesia, 40154
eddypn@upi.edu, rizky_rjp@upi.edu, iman.muhamad@student.upi.edu

Abstract—**Nowadays, it is highly necessary to give precautions aimed to prevent Netizen from creating fake accounts to do criminal offence. One of the means is to use authentication code where it contains activation message. This authentication code is generated through activation message combined with timestamp values which would further be used on One-time Password. Then, it would be encrypted using Advanced Encryption Standard cryptographic algorithm, the generated authentication code can only be used once within a limited time. Unfortunately though, in 2011 three researchers from several universities and Microsoft, Andrey Bogdanov from K.U. Leuven, Dimitri Khovratovich from Microsoft and Christian Reachberger from ENS Paris found a crack on AES encryption, thus requiring modification to enhance AES complexity in order to close the said crack. AES modification can be done on S-Box and ShiftRow, thus enabling their dynamics following the keys given. 256 bits AES is chosen because it contains more key combination and longer time to decrypt compared to other type of AES. After its modification is done, the next step would be tested with Avalanche Effect and Randomness Test, where good cryptographic algorithm would have Avalanche Effect values around 50% and able to pass 5 basic random tests in Randomness Test.**

Keywords—cryptography; advance encryption standard-256 bits; one time password; authentication code; avalanche effect; randomness test;

I. INTRODUCTION

The ever increasing social media and online purchase activities in cyber world has created a phenomenon where there are certain individuals or interest groups use it to benefit themselves or disadvantage the others, by registering fake accounts or bots, thus enabling these irresponsible actors to do improper acts, such as deception, insult, defamation, or falsified support, even black campaign during presidential election.

This obviously harms and gives bad image toward the social media owner or online trading sites. The emergence of rogue and fake accounts. Therefore, preventive measures are necessary in order to minimize those rogue and fake accounts in the social media and online trading sites.

One of the ways is to enhance security measures in registration process by generating authentication code to verify and activate the account.Account activation can implement cryptographic algorithm within the activation message containing authentication code sent through SMS to the mobile phone's number to ensure whether it's bot. This authentication code is in the form of encrypted disposable message and only has rather short lifespan.

Advance Encryption Standard (AES) becomes to most commonly used algorithm in multiple processes and devices. Daemen and Rijmen stated that AES has been through thorough tests on multiple type of security applications [1]. Therefore, Kumar and Karthikeyan claimed that AES can be used in situations where high level of security is required [2].

However in 2011, three researchers from several universities and Microsoft, Andrey Bogdanov from K.U. Leuven, Dimitri Khovratovich from Microsoft, and Christian Reachberger from ENS Paris found flaws in AES encryption as reported by The INQUIRER [3]. It enables probability to solve the secret keys faster than ever.

With the discovery of flaws in AES encryption, this research aims to enhance the complexity of AES, whereby the modification would be based on Shannon principle, which is Confusion and Diffusion. The modification is done on S-Box for Confusion and on ShiftRow for Diffusion.

Next, AES modification will be performed with 256 bits AES because it has more expansion. After modifying the 256 bits AES, then tested using calculation of Avalanche Effect and Randomness Test values because an algorithm is deemed to have good Avalanche Effect if the changes occur approximately around 50% of the previous results, since it is what makes an algorithm to be completely random [4].

In order to give immunity for authentication code generated through AES, it is necessary to add such method and it would use One-time Password. This method makes it possible to give lifespan and validation in each authentication code which can only be used once and has expiration period. One-time Password method is chosen because according to Kim, et al, The aim of One-time Password is to enhance the difficulty in accessing restricted source, illegally [5].

978-1-5090-1721-8/16/$31.00 ©2016 IEEE

Sediyono, etc. suggests that Comparing codes generated by OTP with Pseudo Random Number Generator (PNRG) can make similar code. Under this condition will make it difficult for hacker to decrypt the code in order to trespass and enter into the system [6]. Other than that, Parmar, etc. has also suggested this algorithm is highly economical to implement so that it is available when user is synchronizing [7].

II. Research

A. Advance Encryption Standard (AES) Algorithm

In cryptography, Advance Encryption Standard, or AES is a block cipher used as an encryption standard by the National Institute of Standards and Technology.AES establishes that the key length is of 128, 192 and 256 bits. Therefore, it is then known as AES-128, AES-192 and AES-256. Table 1 summarizes the differences between three versions of the AES.

TABLE I. Three Versions Of The AES [8]

Type of AES	Number of Key (Nk words)	Number of Block (Nb words)	Number of Round (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

To understand the encryption process on AES. Fig. 1 is the structure transformation of AES encryption process.

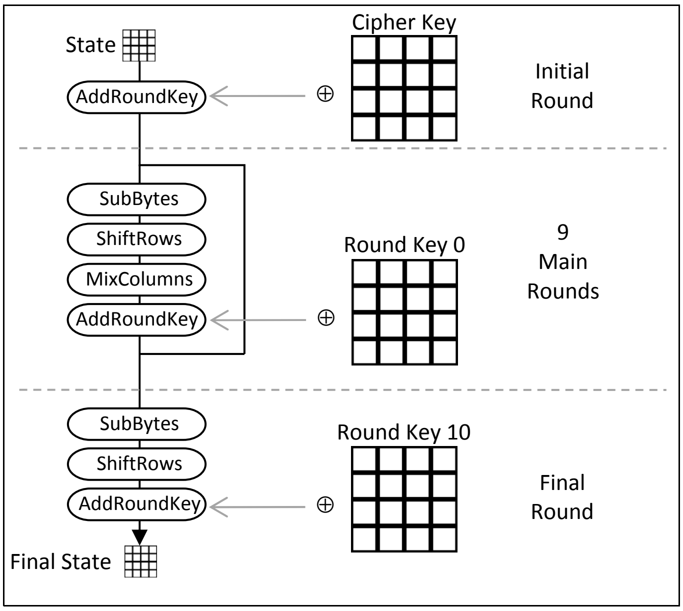


Fig. 1. AES Encryption Process (*AES 128-bits Encryption Process*)

1. *AddRoundKey: perform the XOR operation between the initial state (plaintext) with a cipher key or keys round-0. This phase is also called the initial round.*
2. *SubBytes: every state is replaced with the input in a table S-Box or the substitution box. This operation will give the principle of non-linearity in cipher.*
3. *ShiftRow: perform a leftward shift on a rotating basis every bytes in each row of the state. The number of shifts each byte is different for each line. The first line will remain in its original state. Each byte of the second row is shifted one step to the left. The third and fourth lines shifted to the left as much as two and three steps to the left.*
4. *MixColumns: do randomization of data in each column array state.*

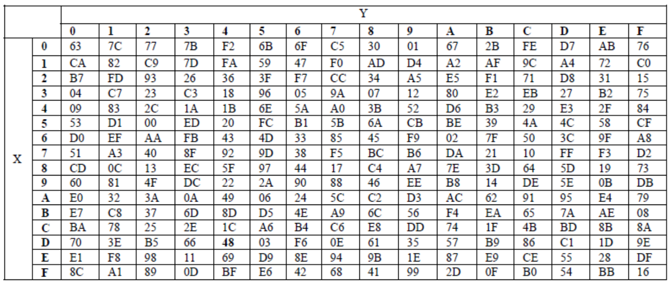
After the encryption process is done as shown in Fig. 1 it will produce cipher text from the plaintext and cipher key.

B. AES 256 Bits Modifications

The modification of 256-bits AES is intended to increase the complexity and increase the immunity of the AES-256 algorithm. Modification would be done on the S-Box and Shift Rows. Modification on S-Box is meant to enhance the confusion of AES and modification on Shift Rows is intended to increase the diffusion of AES.

1) S-Box Modifications

S-Box is an important function that is used on the AES algorithm. In which the S-Box is used in the process of encryption that is on the Sub Bytes and the key expansion process which is on the Sub Word, the function of the S-Box is to produce the non-linear substitute, where each input byte will produce output byte that is independent. Fig. 2 is the commonly used S-Box in AES



		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
X	**0**	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	3B	52	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	**48**	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Fig. 2. AES S-Box [9]

To modify the S-Box, it will depend on the key, so that the S-box produced will dynamically follow the given key, to get the process of S-Box, which dynamically adjusts the key is as follows

1. Perform XOR operation on each of the key elements to obtain XOR key result. As the following examples in Fig. 3

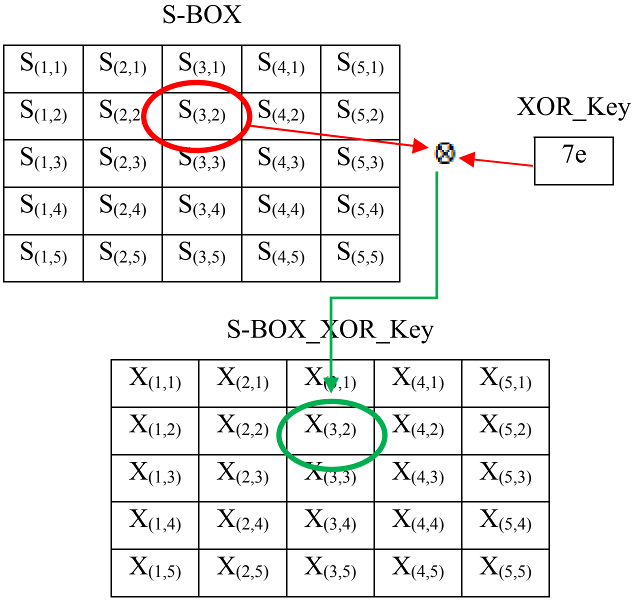
Key : ilmukomputerupi
Key in Hexadecimal : 69 6c 6d 75 6b 6f 6d 70 75 74 65 72 75 70 69 00

$$\boxed{69} \oplus \boxed{6c} \oplus \boxed{6d} \oplus \ldots \oplus \boxed{00} = 7e$$

Result XOR_key : 7e

Fig. 3. Example to find result "XOR_key”

2. After getting the XOR_key, the next step is to perform the XOR operation of each element of the S-Box with the key XOR result and produce S-Box_XOR_key as the following example in Fig. 4.



$S_{(1,1)}$	$S_{(2,1)}$	$S_{(3,1)}$	$S_{(4,1)}$	$S_{(5,1)}$
$S_{(1,2)}$	$S_{(2,2}$	$S_{(3,2)}$	$S_{(4,2)}$	$S_{(5,2)}$
$S_{(1,3)}$	$S_{(2,3)}$	$S_{(3,3)}$	$S_{(4,3)}$	$S_{(5,3)}$
$S_{(1,4)}$	$S_{(2,4)}$	$S_{(3,4)}$	$S_{(4,4)}$	$S_{(5,4)}$
$S_{(1,5)}$	$S_{(2,5)}$	$S_{(3,5)}$	$S_{(4,5)}$	$S_{(5,5)}$

$X_{(1,1)}$	$X_{(2,1)}$	$X_{(3,1)}$	$X_{(4,1)}$	$X_{(5,1)}$
$X_{(1,2)}$	$X_{(2,2)}$	$X_{(3,2)}$	$X_{(4,2)}$	$X_{(5,2)}$
$X_{(1,3)}$	$X_{(2,3)}$	$X_{(3,3)}$	$X_{(4,3)}$	$X_{(5,3)}$
$X_{(1,4)}$	$X_{(2,4)}$	$X_{(3,4)}$	$X_{(4,4)}$	$X_{(5,4)}$
$X_{(1,5)}$	$X_{(2,5)}$	$X_{(3,5)}$	$X_{(4,5)}$	$X_{(5,5)}$

Fig. 4. Example to generate S-Box_XOR_Key

3. The next step is to make Inverse S-Box that depends on the decryption key to be used, it is done through moving the position of each element of the S-Box_XOR_key algorithm as follows in Fig. 5.

```
for i = 1 to 256
        inv_s_box(s_box(i)+1)=i-1;
endfor
```

Fig. 5. Psudocode to find Inverse S-Box

2) *Shift Row Modifications*

Modification on the Shift Rows occurs in the process of shifting every state, which is influenced by the first Round Key

Each line in the first Round Key will undergo XOR operation, and after the results of each XOR row is obtained, there will be a ranking to determine the number of shifts to be performed by each row as the following example in Fig. 6:

Key : ilmukomputer2011
Key in Hexadecimal:
69 6c 6d 75 6b 6f 6d 70 75 74 65 72 32 30 31 31

RoundKey-1			
69	6b	75	32
6c	6f	74	30
6d	6d	65	31
75	70	72	31

XOR each element on the same row

69	⊗	6b	⊗	75	⊗	32	=	45
6c	⊗	6f	⊗	74	⊗	30	=	47
6d	⊗	6d	⊗	65	⊗	31	=	54
75	⊗	70	⊗	72	⊗	31	=	46

Ranking
45 : 1
47 : 3
54 : 4
46 : 2

Fig. 6. Example to determine ranks on each row from the first RoundKey

Having obtained the rank of each row, it can be determined the amount of shift that can be performed by each row in the state by shifting as much ranking- 1, such as on the following example in Fig. 7.

Rank of each row: [1,3,4,2]

State

11	55	99	cc
22	66	00	dd
33	77	aa	ee
44	88	bb	ff

Ranking
1
3
4
2

Shifting on State

11	55	99	cc
00	dd	22	66
Ee	33	77	aa
88	bb	ff	44

Fig. 7. The shift on each line in accordance to its ranking

By doing the above process we would obtain Shift Rows to shift to the right and Inverse Shift Rows to shift to the left which is dynamic depends on the key.

C. *One Time Password Method*

One time password concept originated from the need to do periodic password replacement in order for the password to stay safe and not misused, password replacement is done automatically.

The generated passwords cannot be reused (non-reusable), so it's useless if someone finds their password, because the password will expire. Each password is used only once [10]

OTP provides immunity and security from possible password sniffing attacks, even if someone peek your password, they still cannot get access to your account.

Here is how One Time Password works and the example in table II:

1. Taking the value of timestamp when OTP is generated.
2. Combining value of OTP unit time with a secret message to be encrypted
3. Encrypt the combined result of the value of the unit of OTP time with a secret message
4. Toke some initial digits from encryption to be used as a One Time Password

TABLE II. EXAMPLE OF ONE TIME PASSWORD

Message	Timestamp OTP	Combine	MD5(Combine)	OTP
Hallo	12485046	hallo12485 046	b6bd541f6198900ce 703379	b6bd5
Hallo	12485047	hallo12485 047	aa94894ff7c66e9a8b 24f95d	aa948
Hallo	12485048	hallo12485 048	b1cdb9995454a2bd7 316a645	b1cdb
Hallo	12485049	hallo12485 049	db2ae1c7e4c2de3c1 be7783f4	db2ae

D. *Steps to Encrypt and Decrypt Authentication Code*

In making the authentication codes it is needed to know how to generate or dismantle it, therefore, there is needs to design how to call upon and unload authentication code, so that the system will be running as planned, Fig. 8 describes the flow to generate and decode authentication.

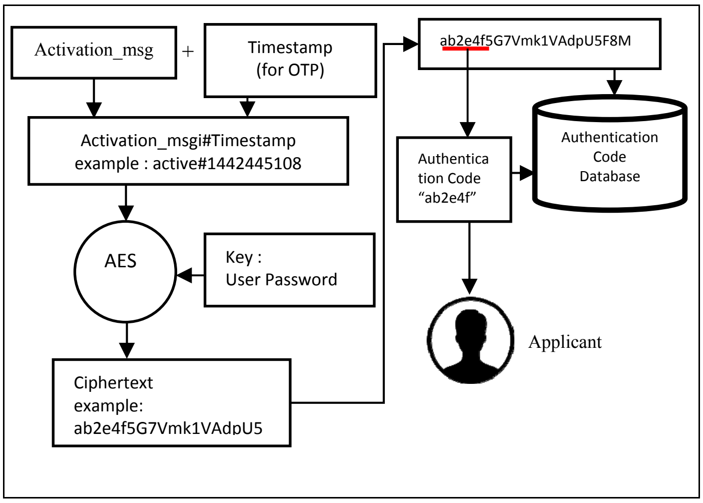


Fig. 8. Generating Authentication Code

This is the flow to generate the authentication codes.

1. Set up activation message and the value of the unit time (timestamp) then combine both into plaintext.
2. Having obtained plaintext, the next step is to put it into AES 256 encryption key that has been given in the form of applicant's passwords.
3. After going through the encryption process, the cipher text will be obtained.
4. Then the first 6 digits of cipher text produced will be taken to be used as the authentication code, then the cipher text and authentication code will be stored in the database.
5. The next step is to send the authentication code to the registrar to be entered on the stage of activation of a new account

After Applicant receive authentication code, Applicant can use that code to activate their account. Fig. 9 is the scheme how to decrypt Authentication Code.

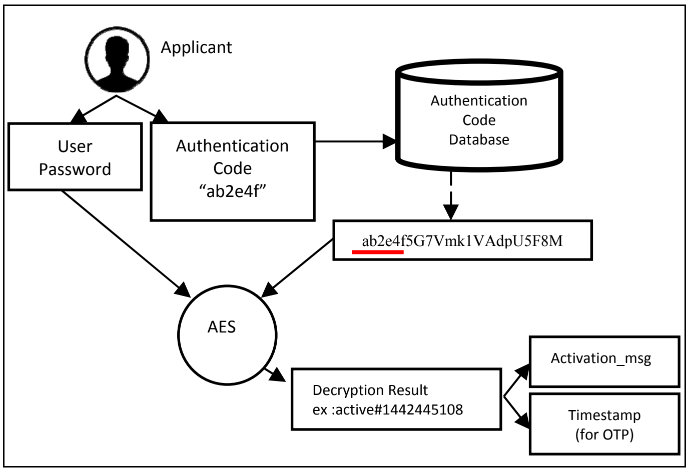


Fig. 9. Decrypt Authentication Code

The flow to dismantle the authentication code that has been given to the applicant is as follows.

1. Applicants enter the authentication code received along with the registered password
2. The authentication code entered by applicant would be looked out cipher text pair on the database of authentication code.
3. After the cipher text is found then the next step is the process of decryption keys belonging to the applicant in the form of a password.
4. Then the results obtained from the decrypted cipher text is plaintext
5. Plaintext is then separated between an activation message to the value of the unit of time to be calculated whether the authentication code is valid

E. *Process to Send Authentication Code*

Authentication code sent via SMS to the mobile phone number belonging to the applicant. Fig. 10 is flow of an authentication code delivery process that uses the services of third-party SMS Gateway.

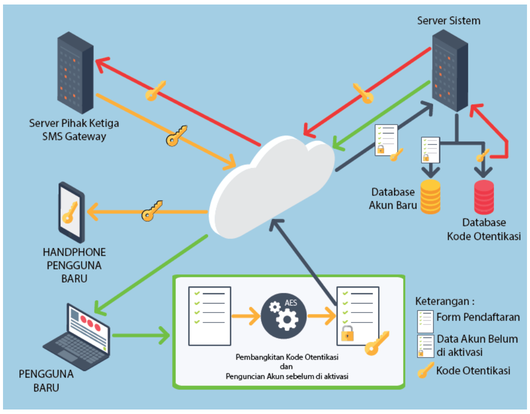
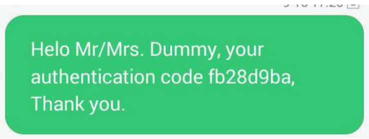


Fig. 10. Process to Send Authentication Code

Fig. 11 is a sample SMS authentication code that is sent to the mobile phone numbers of applicants.



Helo Mr/Mrs. Dummy, your authentication code fb28d9ba, Thank you.

Fig. 11. Sample of Authentication Code

III. AES 256 Bits Modifications Test Result

A. *Avalanche Effect(AE)*

The testing procedure upon the value of the avalanche effect is very important for cryptographic algorithms. It can be seen the results of different encryption of plaintext or key that experience one bit change. An algorithm is stated to have good AE good if changes that occur is about 50% of the previous results, because this is what makes a truly random algorithm [4].

If a cipher block does not produce an avalanche effect with significant increase, there will only be a slight shuffling in the process of making the cipher text, and thus cryptanalyst can make predictions about the input or plaintext to capitalize output or cipher text only. Output obtained may be enough to solve most or the entire algorithm. The condition is certainly not desirable from the point of view of the designer of cryptographic algorithm [11].

Table III are the results of the calculation of the value of the avalanche effect with plaintext "11 22 33 44 55 66 77 88 99 00 aa bb cc dd ee ff".

TABLE III. RESULT OF AVALANCHE EFFECT TEST

AES 256 *bits* Name	Key Pair (K)		Encryption Result	*Avalanche Effect* Value
Standard	K1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8e 05 c2 b8 17 aa 3a 0a 2e a5 cd 34 55 22 5e f2	48.44%
	K2	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80 80 0a 63 bc a7 ba e0 11 31 9b 53 39 48 77 31	
S-Box &Shift Rows Depending on the key	K1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	8e 05 c2 b8 17 aa 3a 0a 2e a5 cd 34 55 22 5e f2	52.34%
	K2	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	64 e2 98 cf 2b 38 f4 8a 3c 72 b8 79 01 7c d1 74	
Standard	K1	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	4b c5 be ad 87 70 85 9e 4c 84 4c 33 ad 37 32 75	53.90%
	K2	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	d1 68 44 a2 12 c9 cc 9f ea b2 ab c4 80 39 9e 81	
S-Box &Shift Rows Depending on the key	K1	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	4b c5 be ad 87 70 85 9e 4c 84 4c 33 ad 37 32 75	48.44%
	K2	11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11	0f 5f 4a 2c 5a b0 94 55 d7 7a 42 ef b1 29 82 29	

Based on Table III test results, it can be seen that 256 *bits* AES that have been modified is producing *Avalanche Effect* value around 50%

B. *Randomness Test*

To test the randomness of the encryption results, then there is a test conducted, consisting of five basic tests for randomness which will carry out 5 pieces of statistical tests that are typically used to determine whether a row binary characteristics such as directed by a line that is totally random [10].

Testing is done with the help of Cryptool 1.4.3. *software*. Here are the results comparing standard AES 256 *bits* with AES 256 *bits* that have been modified. And Table IV is the result of Randomness Test

TABLE IV. RESULT OF RANDOMNESS TEST

Key		10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
Randomness Test		AES 256 *bits* Algorithm			
Type	Max. Value	Standard		S-Box &Shift Rows Depend Key	
Frequency	6,63	2.53	Pass	0.125	Pass
Poker	18.4	3.71	Pass	2.19	Pass
Run	13.8	3.39	Pass	5.84	Pass
Long Run	34	11	Pass	7	Pass
Serial	9.21	3.51	Pass	1.78	Pass
Key		11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11			
Randomness Test		AES 256 *bits* Algorithm			
Type	Max. Value	Standard		S-Box &Shift Rows Depend Key	
Frequency	6,63	1.53	Pass	0.125	Pass
Poker	18.4	5.61	Pass	4.09	Pass
Run	13.8	5.71	Pass	8.93	Pass
Long Run	34	9	Pass	5	Pass
Serial	9.21	2.71	Pass	1.81	Pass

C. *Execute Time Complexity*

Table VI shows the required time complexity to encrypt plaintext and key in the Table V using 256 bits AES original and with modified 256 bits AES.

TABLE V. PLAINTEXT AND KEY

Plaintext	ilmu komputer
Key	universitas pendidikan indonesia

TABLE VI. RESULT OF TIME COMPLEXITY

Ciphertext AES 256 bits		Time for Encryption
Standard	5b2434cdd83c93b1779 f0a1cf0398746	0.354 s
Modifications	3891b2c62e94812565a0 55b66e05da12	0.511 s

It takes more time to execute AES 256 bits Modifications then AES 256 bits Standard because Its complexity to generate new S-Box that depending on Key and ShiftRow that depending on the first round of key expansion.

IV. CONCLUSION

The conclusion of this paper is the use of authentication code to activate new accounts to minimize the account creation of rogue or fake accounts, in which the authentication code is derived from the activation message and the encrypted value timestamp for OTP by using a cryptographic algorithm.

Cryptographic algorithm AES-256 *bits* that have been modified with S-Box and Shift Rows has passed testing avalanche effect and randomness tests can be implemented to generate the authentication code, this algorithm takes more time to execute than the standard AES-256 algorithm

REFERENCES

[1] Joan Daemen and Vincent Rijmen, "Rijndael, The Advance Encryption Standard," *Dr. Dobb's Journal*, vol. 3, pp. 137-139, 2001.

[2] M. Kumar and D. Karthikeyan, "Investigating Efficiency of Blowfish and Rijndael (AES) Algorithm," *I. J. Computer Network and Information Security*, pp. 22-28, 2012.

[3] Dave Neal. (2011, August) The Inquirer. [Online]. http://www.theinquirer.net/inquirer/news/2102435/aes-encryption-cracked

[4] A. Rohan, *Advance Encryption Standard (AES) Modes of Operation*. Maryland: University of Maryland, 2011.

[5] M. J. Kim, B. H. Lee, and S. J. Kim, "Weakness and Improvments of a One-time Password Authentication Scheme," *International Journal of Future Generation Communication and Networking*, vol. 2, no. 4, pp. 29-39, 2009.

[6] E. Sediyono, K. I. Santoso, and Suhartono, "Secure Login by Using One-time Password Authentication Based on MD5 Hash Encrypted SMS," *International Conference on Advances in Computing, Communication and Informatic*, pp. 1604-1608, 2013.

[7] H. Parmar, N. Nainan, and S. Thaseen, "Generation of Secure One-time Password Based on Image Authentication," *Computer Science & Information Technology*, pp. 195-206, 2012.

[8] M. F. Wali and M. Rehan, *Effective Coding and Performance Evaluation of The Rijndael Algorithm (AES)*. Karachi: NED University of Engineering and Technology, 2005.

[9] William Stalling, *Cryptograhpy and Network Security Principles and Practices*, 5th ed. New Jersey, United States of America: Pearson Education, 2011.

[10] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton: CRC Press, 1996.

[11] G. Krishnamurthy and D. Ramaswamy, "Making AES Stronger: AES with Key Dependent S-Box," *International Journal of Computer Science and Network Security*, vol. 8, no. 9, pp. 388-398, 2008.