Journal of Soft Computing Exploration

# Data Security System of Text Messaging Based on Android Mobile Devices Using Advanced Encrytion Standard Dynamic S-BOX

**Akhmad Sahal Mabruri [1], Alamsyah[2]**
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

**Article Info**

***Article history:***

Received Jul 29, 2020
Revised Aug 12, 2020
Accepted Sept 3, 2020

***Keywords:***

Android
Chat Messaging
Cryptography
Advanced Encrytion Standard
Dynamic S-BOX

**ABSTRACT**

Most of the recent technologies are turning to mobile platforms, Android becames one of the most widely used OS. Eventhough it has complete features, even it's not safe enough such like Chat Messenger. The security of messages distribution is a challenge to increase of vulnerable distribution of information through the network today. Therefore, a data security or cryptographic algorithm is needed to secure the messages so that it cannot be read by irresponsible people. National Institute of Standard and Technology (NIST) established the Advanced Encrytion Standard (AES) cryptographic algorithm as a standard encryption algorithm that is safe and can be used globally. AES algorithm is included in block cipher cryptography that uses substitution boxes (S-BOX) in its operations, so that algorithmically can make input and output unrelated. So, it can provide more varied output in the process, we need a dynamic S-BOX. In this research, dynamic S-BOX generalized using XOR operations from affine transformations with 8-bit binary element matrices arranged and randomly to produce as many as 256 S-Boxes. The application of dynamic AES with S-BOX algorithm on Android-based messenger chat application is built using the Java programming language and database hierarchy for data storage. The implementation results showed that the algorithm was running well and could encrypt the text of the message to ciphertext and decrypt the ciphertext to the original message. This research can be used as a reference so that further researchers can merge the AES algorithm with other algorithms to improve the security of encryption in text files, documents, images, videos or other types of files.



***Corresponding Author:***

Akhmad Sahal Mabruri
Computer Science Departement
Faculty of Mathematich and Natural Sciences, Universitas Negeri Semarang,
Email: mabroery7@gmail.com

## 1. INTRODUCTION

The security of information distribution is an important aspect. The delivery of information requires a process that can ensure the security and data integrity [1]. One of study that can be implemented is cryptography. Cryptography came from the Greek language, which is from the terms of crypto which means secret and graphia which means writting. Cryptography is a science or art used to maintain the security of messages when messages were sent from one place to another. The term well known as encryption and decryption [2]. In cryptography, there is an algorithm that became encryption standards set by the National Institute of Standard and Technology (NIST) in October 2000, namely the Advanced Encryption Standard (AES). AES is included in the type of cryptographic algorithms which symmetrical and cipher block characteristics [3]. This algorithm

uses the same key when encryption and decryption and the input and output in the form of blocks with a certain number of bits. The bit size in AES can be 128-bit, 192-bit or 256-bit, named AES-128, AES-192, and AES-256 [4]. The selection of the size of the data block and key will determine the number of processes that must be passed for the encryption and decryption process. However, de-facto there are only two variants of AES, namely AES-128 and AES-256, because users will rarely use 192-bit long keys [5]. Because AES has at least 128-bit key lengths, AES is resistant to exhaustive key search attacks with current technology. Using a 128- bit key length, there are $2^{128} = 3{,}4 \times 10^{38}$ possible keys.

AES is included in block cipher encryption, like other block cipher encryption, AES also uses substitution boxes to replace inputs and outputs to be unrelated. AES has four transformation data for Substitution bytes (SubBytes) / Inverse SubBytes (Inv SubBytes), ShiftRow / Inverse ShiftRow (Inv ShiftRow), MixColumn / Inverse MixColumn (Inv MixColumn) and AddRoundkey. [6]. But unlike other block cipher algorithms that use dynamic S-BOX, AES specifies a static S-BOX that is used in the encryption and decryption process. S-BOX has an important role in the implementation of AES. S-BOX is used to randomize input bits that will produce output bits [7].

This study discussed about how to design and implement the dynamic S-BOX Advance Encryption Standard (AES) method for the security of text message data that will be sent and received by the user. So, it will be designed a chat or messaging application that implements AES cryptographic algorithm using dynamic S-BOX. The purpose of this study is securing data and generating text message encryption applications on chat messengers by using dynamic S-Box AES algorithm method.

## 2. METHOD

This research was built by developing AES method and Dynamic S-Box into the systems. The system was developed by waterwall model which consist five stages. The waterfall model can be seen in Figure 1. Communication stage in the waterfall model of a software or system to be built requires analysis of software requirements in the form of collecting additional data in journals, articles, books, etc. Furthermore, in the planning stage, namely the plan for implementing a generated dynamic S-Box on the AES-128 bit cryptographic algorithm. So, it produces a user requirement document or in other words data that relates to the user's desire in developing the software. The next stage that must be passed is modeling, this process will translate the requirements to a software design. This process focuses on data structure design, software architecture and algorithmic / procedural details. After modeling, the next stage is construction or developing the system using the coding process. After the coding is complete, testing of the system will be done. The goal is that when testing can be found errors in the system can be corrected. The last stage is deployment, in this process making the system is in the final stage. Then the system that has been made must be regularly maintained.

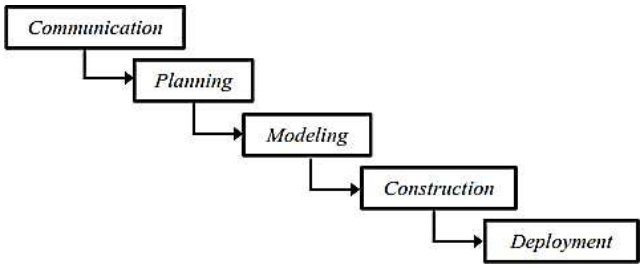


Figure 1. The waterfall model

### 2.1 Advanced Encrytion Standard

Advanced Encrytion Standard (AES) is one of cryptography algorithm which symmetry and block cipher. This algorithm uses the same key to encrypt and decrypt as well as input and output in the form of blocks with a certain number of bits [7]. AES supports a variety of block sizes and keys that will be used. However, Rijndael has a fixed block and key size of 128, 192, 256 bits. The selection of block size and key data will determine the number of processes that must be passed for the encryption and decryption process. This encryption technique includes the type of block cipher as well as DES. The main difference between the AES encryption technique and the DES encryption technique is that AES uses substitutions or often called S-boxes

### 2.2 Dynamic S-BOX

S-Box is a matrix that contains simple substitutions which mapped one or more bits with one or more other bits. It's also well known as most cipher block algorithms. S- Box mappped m input bits into n output bits, so that the S-Box is called m × n S-Box [8]. Substitution process that mappped input based on look-up tables. Usually the input from the operation on the S-Box is used as the index and the output is the entry.

### 2.3 Android

Android is an operating system for mobile phones based on Linux. Android is a computer code-based software that can be distributed openly or well know as open source so that programmers can create new applications in it [9]. There is an Android Market that provides thousands of applications which free and paid, and has a native Google application that is integrated, such as push email GMail, Google Maps, and Google Calendar. The android application was developed with the Java programming language, using the Android software development kit (SDK). This SDK consists of a set of tools for development, including a debugger and software library, a QEMU-based handset emulator, documentation, sample code and tutorials.

### 2.4 Firebase Real Time Database

Firebase Realtime Database is a NoSQL cloud-hosted database service which owned by Firebase SDK. This service offers data storage that can be synchronized in real time to all connected clients and can support flick mode for situations when an internet connection is not available [10]. Firebase in the backend has several components, namely Cloud messaging, Authentication, Realtime, Storage, and Hosting. Cloud messaging can be used to send messages between users. First, authentication simplifies application integration by limiting user access. Second, firebase can be used in realtime or offline. Third, the storage also allows to keep image, audio and video content. The last component is that hosting which is possible to be developed globally [11].

### 2.5 Chat Messenger

Chat messenger is one of technology that provides a feature of communication between two or more people using a network that allows users to send and receive messages in real time [12]. In addition, chat messenger is also called an instant messaging application or instant message on a computer network technology that allows users to send messages to other users who are connected on a computer or internet network. Communication in chat is generally in the form of text (text chat).

## 3. RESULT AND DISCUSSION

In this research the implementation of AES algorithm in chat applications developed using application namely Android Studio. The development used the waterfall method which consists of Analysis of Requirement, Design, Implementation and Testing. Analysis of Requirement was conducting by preparing all the needs in the developing of application. The design stage was conducted by making the application interface and databases. The implementation stage was conducted by coding and applying AES using dynamic S-BOX. The testing stage was carried out using the Black Box method. The stages of 128-bit AES algorithm encryption in securing text messages are as follows:

The encryption stages with AES algorithm using Dynamic S-BOX:

| | |
|---|---|
| Plaintext: | selamat pagi pak |
| In Hexa: | 73 65 6C 61 6D 61 74 20 70 61 67 69 20 70 61 6B |
| Key: | kunciku |
| In Hexa: | 6B 75 6E 63 69 6B 75 00 00 00 00 00 00 00 00 00 |

**Step 1:** Continue the calculation, prepare two 4x4 matrices from plaintext and key

$$\text{Plaintext (Hex)} \quad : \begin{bmatrix} 73 & 6D & 70 & 20 \\ 65 & 61 & 61 & 70 \\ 6C & 74 & 67 & 61 \\ 61 & 20 & 69 & 6B \end{bmatrix}$$

Key (Hex) : $\begin{bmatrix} 6B & 69 & 00 & 00 \\ 75 & 6B & 00 & 00 \\ 6E & 75 & 00 & 00 \\ 63 & 00 & 00 & 00 \end{bmatrix}$

**Step 2:** Conduct XOR on plaintext with roundkey. The XOR process between the corresponding columns of the two matrices begins by switching the hexadecimal data of each column into binary form. This step is called addroundkey, which will generate a new matrix.

$$\begin{bmatrix} 73 & 6D & 70 & 20 \\ 65 & 61 & 61 & 70 \\ 6C & 74 & 67 & 61 \\ 61 & 20 & 69 & 6B \end{bmatrix} \text{XOR} \begin{bmatrix} 6B & 69 & 00 & 00 \\ 75 & 6B & 00 & 00 \\ 6E & 75 & 00 & 00 \\ 63 & 00 & 00 & 00 \end{bmatrix}$$

Binner Hex 73 XOR 6B= 0111 0011 XOR 0110 1011= 0001 1000 = 18
Binner Hex 65 XOR 75= 0110 0101 XOR 0111 0101 = 0001 0000 = 10
Binner Hex 6C XOR 6E= 0110 1100 XOR 0110 1110 = 0000 0010 = 02
Binner Hex 61 XOR 63= 0110 0001 XOR 0110 0011 = 0000 0010 = 02
Binner Hex 6D XOR 69= 0110 1101 XOR 0110 1001= 0000 0100 = 04
Binner Hex 61 XOR 6B= 0110 0001 XOR 0110 1011= 0000 1010 = 0A
Binner Hex 74 XOR 75= 0111 0100 XOR 0111 0011 = 0000 0111 = 07
Binner Hex 20 XOR 00= 0010 0000 XOR 0000 0000 = 0010 0000 = 20
Binner Hex 70 XOR 00= 0111 0000 XOR 0000 0000 = 0111 0000 = 70
Binner Hex 61 XOR 00= 0110 0001 XOR 0000 0000 = 0110 0001 = 61
Binner Hex 67 XOR 00= 0110 0111 XOR 0000 0000 = 0110 0111 = 67
Binner Hex 69 XOR 00= 0110 1001 XOR 0000 0000 = 0110 1001 = 69
Binner Hex 20 XOR 00= 0010 0000 XOR 0000 0000 = 0010 0000 = 20
Binner Hex 70 XOR 00= 0111 0000 XOR 0000 0000 = 0111 0000 = 70
Binner Hex 61 XOR 00= 0110 0001 XOR 0000 0000 = 0110 0001 = 61
Binner Hex 6B XOR 00= 0110 1011 XOR 0000 0000 = 0110 1011 = 6B

So as producing a matrix, as follows:

$$\begin{bmatrix} 18 & 04 & 70 & 20 \\ 10 & 0A & 61 & 70 \\ 02 & 01 & 67 & 61 \\ 02 & 20 & 69 & 6B \end{bmatrix}$$

**Step 3:** After getting the XOR result matrix between plaintext and roundkey, the substitution process is conducted with dynamic S-BOX, which the dynamic S- BOX table on the keyword "kunciku" is shown in Table 1.

Tabel 1. Dynamic S-BOX Table on the Keyword “kunciku”

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **a** | **b** | **c** | **d** | **e** | **f** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 8 | 23 | 28 | 16 | 153 | 0 | 4 | 174 | 91 | 106 | 12 | 64 | 149 | 188 | 192 | 29 |
| **1** | 161 | 233 | 162 | 22 | 145 | 50 | 44 | 155 | 198 | 191 | 201 | 196 | 247 | 207 | 25 | 171 |
| **2** | 220 | 150 | 248 | 77 | 93 | 84 | 156 | 167 | 95 | 206 | 142 | 154 | 26 | 179 | 90 | 126 |
| **3** | 111 | 172 | 72 | 168 | 115 | 253 | 110 | 241 | 108 | 121 | 235 | 137 | 128 | 76 | 217 | 30 |
| **4** | 98 | 232 | 71 | 113 | 112 | 5 | 49 | 203 | 57 | 80 | 189 | 216 | 66 | 136 | 68 | 239 |
| **5** | 56 | 186 | 107 | 134 | 75 | 151 | 218 | 48 | 1 | 160 | 213 | 82 | 33 | 39 | 51 | 164 |
| **6** | 187 | 132 | 193 | 144 | 40 | 38 | 88 | 238 | 46 | 146 | 105 | 20 | 59 | 87 | 244 | 195 |
| **7** | 58 | 200 | 43 | 228 | 249 | 246 | 83 | 158 | 215 | 221 | 177 | 74 | 123 | 148 | 152 | 185 |
| **8** | 166 | 103 | 120 | 135 | 52 | 252 | 47 | 124 | 175 | 204 | 21 | 86 | 15 | 54 | 114 | 24 |
| **9** | 11 | 234 | 36 | 183 | 73 | 65 | 251 | 227 | 45 | 133 | 211 | 127 | 181 | 53 | 96 | 176 |
| **a** | 139 | 89 | 81 | 97 | 34 | 109 | 79 | 55 | 169 | 184 | 199 | 9 | 250 | 254 | 143 | 18 |
| **b** | 140 | 163 | 92 | 6 | 230 | 190 | 37 | 194 | 7 | 61 | 159 | 129 | 14 | 17 | 197 | 99 |
| **c** | 209 | 19 | 78 | 69 | 119 | 205 | 223 | 173 | 131 | 182 | 31 | 116 | 32 | 214 | 224 | 225 |
| **d** | 27 | 85 | 222 | 13 | 35 | 104 | 157 | 101 | 10 | 94 | 60 | 210 | 237 | 170 | 118 | 245 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **e** | 138 | 147 | 243 | 122 | 2 | 178 | 229 | 255 | 240 | 117 | 236 | 130 | 165 | 62 | 67 | 180 |
| **f** | 231 | 202 | 226 | 102 | 212 | 141 | 41 | 3 | 42 | 242 | 70 | 100 | 219 | 63 | 208 | 125 |

The dynamic S-Box in this research was reshaped based on the modification of the affine transformation. The dynamic S-Box table has the same size as the original AES S-box table, but in that dynamic S-box there are 256 S-Boxes and the work process is constantly changing and randomly. In this research, we would create a generalized S-Box table that uses generalized XOR operations from affine transformations with 8-bit matrix binary elements arranged randomly to produce 256 different matrix shapes.

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Substitution Results from Dynamic S-BOX, as follows:

$$\begin{bmatrix} C6 & 99 & 3A & DC \\ A1 & 0C & 84 & 3A \\ 1C & 17 & EE & 84 \\ 1C & DC & 92 & 14 \end{bmatrix}$$

**Step 4:** In the substitution results with the new S-BOX, the shiftrows have been carried out. The shiftrows process is very simple by doing wrapping (cyclic).

$$\begin{bmatrix} C6 & 99 & 3A & DC \\ 0C & 84 & 3A & A1 \\ EE & 84 & 1C & 17 \\ 14 & 1C & DC & 92 \end{bmatrix}$$

**Step 5:** After the results of the shiftrows was obtained, then mixcoloumn is conducted by multiplying the result matrix shiftrows with the rijndael matrix. This transformation is expressed as matrix multiplication. All plaintexts are carried out the same process so that the new mixcoloumns matrix will be obtained.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} C6 \\ 0C \\ EE \\ 14 \end{bmatrix} = \begin{bmatrix} 79 \\ E3 \\ 31 \\ 9B \end{bmatrix}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 99 \\ 84 \\ 84 \\ 1C \end{bmatrix} = \begin{bmatrix} 26 \\ 01 \\ 2A \\ 88 \end{bmatrix}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 3A \\ 3A \\ 1C \\ DC \end{bmatrix} = \begin{bmatrix} FA \\ B6 \\ 47 \\ CB \end{bmatrix}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} DC \\ A1 \\ 17 \\ 92 \end{bmatrix} = \begin{bmatrix} DE \\ 2E \\ FE \\ F6 \end{bmatrix}$$

So as to produce the mixcoloumns matrix as follows:

$$\begin{bmatrix} 79 & 26 & FA & DE \\ E3 & 01 & B6 & 2E \\ 31 & 2A & 47 & FE \\ 9B & 88 & CB & F6 \end{bmatrix}$$

**Step 6:** Conduct XOR between the matrix of mixcoloumns with the key generated from the key expansion process. Repeat the 3rd to 7th steps for the 1st to 9th iterations because this study uses AES-128. The results of the AES algorithm encryption calculation use dynamic S-BOX, which is as follows:

addRoundKey(1) 18100202040A01207061676920706l6B
subBytes(1) C6A11C1C990C17DC3A84EE92DC3A8414
shiftRows(1) C60CEE149984841C3A3A1CDCDCA11792
mixColumns(1) 79E3319B26012A88FAB647CBDE2EFEF6
-
-
-
addRoundKey(9) A720A27BF7DC51C4C25F76F874C18D1D
subBytes(9) 37DC514A03EDBA774EA4532AF91336CF
shiftRows(9) 37ED53CF03A4364A4E135177F9DCBA2A
mixColumns(9) DECC36628D4015038FEC667E06A53422
addRoundKey(10)51B95801C15E0E600FF27D1D0EBB2F41
subBytes(10) BA3D01171333C0BB1DE294CFC0817EE8
shiftRows(10) BA3394E813E27E171D8101BBC03DC0CF
RoundKeyAkhir 0346FA8BE6890B1768F46FD8BD56B5CF
Output 0346FA8BE6890B1768F46FD8BD56B5CF

The algorithm is run every time you send or open a conversation to be able to read the message. Then, the figure of application itself can be seen in Figure 2 which is the list of conversations or chat rooms on the application.

When choosing a chat room, the user is required to fill the encryption key on the application as a key to read the existing message and as a key to encrypt the message to be sent. For interfaces when key input can be seen in Figure 3.


Figure 2. *Chat Room Interface*


Figure 3. *Input Encrypt Key*

In the chat interface, the message text can be seen if the key entered was the same as the key used when sending the message, and if the key entered was different then the message can not be read. The interface of the chat room interface that was successfully can read seen in Figure 4, and the interface of the chat room opened with a different key can be seen in Figure 5.


Figure 4. Chat Room Interface with Correctly Key

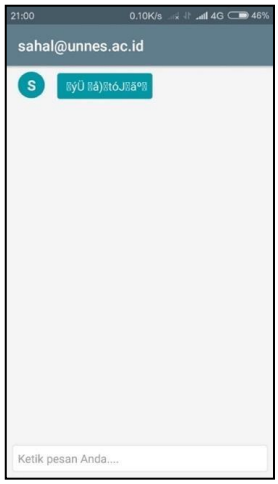


Figure 5. Chat Room Interface with Incorrectly Key

## 4. CONCLUSION

The developing of this application was made using the Android Studio framework with the java programming language and using hierarchical databases. There were 4 stages that must be conducted, namely, analysis of requirement to determine the needs of users and systems to the application, the design stage to design the appearance and process of the application, system design that has been done before, the implementation stage to build an application based on the results of the analysis and testing conducted black box method. The dynamic S-BOX of AES chat application could run properly. In the form of text messages that encrypt the plain text into ciphertext. AES dynamic S-BOX chat application had also managed to decrypt the changed text (ciphertext) as before by using the same key when opening the received message.

## REFERENCES

[1] M.A. Muslim, B. Prasetiyo, and Alamsyah, "Implementation Twofish Algorithm for Data Security in A Communication Network Using Library Chilkat Encryption Activex, " *Journal of Theoretical & Applied Information Technology*, vol. 84, no. 3, pp. 370-375, 2016.

[2] R. Venkateswaran, and V. Sundaram, "Information Security: Text Encryption and Decryption with Poly Substitution Method and Combining the Features of Cryptography, " *International Journal of Computer Applications*, vol. 3, no. (7), pp. 28-31, 2010

[3] C. Sanchez-Avila, and R. Sanchez-Reillol, The Rijndael Block Cipher (AES Proposal): A Comparison With DES. Proceedings of 35th *International Carnahan Conference IEEE.* London, England, Okotber 16, 2001.

[4] M. Rao, T. Newe, and I. Grout, AES Implementation on Xilinx FPGAs Suitable for FPGA Based WBSNs. Proceedings of 9th International Conference Sensing Technology (ICST) IEEE. Massey, New Zealand, December 8 2015.

[5] O. Dunkelman, N. Keller, and A. Shamir, "Improved Single-Key Attacks on 8-Round AES-192 and AES-256., " *Journal of Cryptology*, vol. 28, no. 3, pp. 397-422, 2015

[6] H. Hamzah, N. Ahmad, M.H. Jabbar and C.F. Soon, "AES S-Box/Inv S- Box Optimization Using FPGA Implementation, " *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* vol. 9, no. 3, pp. 133-136. 2017

[7] Alamysah, A. Bejo, A. and Adji, T.B. (2017). *AES S-Box Construction Using Different Irreducible Polynomial and Constant 8-bit Vector*. Proceedings of *IEEE Conference on Dependable and Secure Computing*. Taipe, Taiwan, August, 7, 2017.

[8] U. Çavuşoğlu, S. Kaçar , I. Pehlivan, and A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-Box."*Chaos, Solitons & Fractals,* vol. 95, pp. 92-101. 2017.

[9] B. Shebaro, O. Oluwatimi, and E. Bertino, "Context-based access control systems for mobile devices, " *IEEE Transactions on Dependable and Secure Computing.* vol. 12, no. 2, pp.150-163, 2015.

[10] Alepis, E. and Nita, S. (2017). Mobile Application Providing Accessible Routes for People with Mobility Impairments. Proceeding of *8th International Conference Information, Intelligence, Systems & Applications (IISA) IEEE*. Lanarca, Cyprus, August, 27, 2017.

[11] M.A. Mohammed, A.S. Bright, C. Apostolic, F.D. Ashigbe, and C. Somuah, "Mobile-Based Medical Health Application-Medi-Chat App, " *International Journal of Scientific & Technology Research,* vol. 4, no. 8, pp. 70-76, 2015.
[12] R. Sanjaya, and A. S. Girsang, Implementation Application Internal Chat Messenger Using Android System. Proceeding of International Conference in Applied Computer and Communication Technologies (ComCom*),* Jakarta, Indonesia, May, 17, 2017.