# 12

*by* A L

**Submission date:** 11-Jun-2021 04:00PM (UTC+0700)
**Submission ID:** 1604610156
**File name:** 17589-47062-1-PB.pdf (579.69K)
**Word count:** 4264
**Character count:** 21760

Scientific Journal of Informatics
Vol. 6, No. 1, May 2019
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

# Security Login System on Mobile Application with Implementation of Advanced Encryption Standard (AES) using 3 Keys Variation 128-bit, 192-bit, and 256-bit

**Hamdan Dian[1], Riza Arifudin[2], Alamsyah[3]**

[1,2,3]Computer Science Departement, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Semarang, Indonesia
Email: [1]hamdandianrhu@gmail.com, [2]riza.arifudin@gmail.com, [3]alamsyah@mail.unnes.ac.id

**Abstract**

The development of mobile applications is unbalanced with the level of its security which is vulnerable of hacker attacks. Some important things that need to be considered in the security of mobile applications are login and database system. A login system that used the database as user authentication and passwords are very vulnerable to be hacking. In securing data, various ways had been developed including cryptography. Cryptographic algorithms used in securing passwords usually used MD5 encryption. However, MD5 as a broader encryption technique is very risky. Therefore, the level of login system security in an android application is needed to embed the Advanced Encryption Standard (AES) algorithm in its process. The AES algorithm was applied using variations of 3 keys 128-bit, 192-bit, and 256-bit. Security level testing was also conducted by using 40 SQL Injection samples which the system logins without security obtained 27.5% that be able to enter the system compared to the result of login systems that use AES algorithm 128-bit, 192-bit or 256-bit was obtained 100% that cannot enter into the system. The estimation of the average encryption process of AES 128, 192 and 256 bits are 5.8 seconds, 7.74 seconds, and 9.46 seconds.

**Keywords**: Cryptography, Advanced Encryption Standard, Login System, Mobile Application, Android

## 1. INTRODUCTION

The use of smartphones are increasing and indicating that civilization today has entered to technological era. There are many advanced features that can be found on a smartphone, so that it can replace the mobile phone that used to be an important item for the community. Based on reports from the organizers of the Internet Retailing Expo stated that the majority of internet users in Indonesia use mobile phones (Smartphones) to access the web in 2014. These devices are also mostly used for online shopping.

In 2017 there were many cases of hackers who had hacked many electronic devices especially on mobile applications. There were also many of hacking tools on Android had indicated that mobile applications impacted to security that very vulnerable to

hacker attacks. It is often assumed that system security in mobile applications is also one of the causes that makes it very vulnerable to be hacked.

Some important things that need to be considered in the security of mobile applications and become a problem of vulnerabilities are login and database system. A login system that use the database as user authentication and passwords is very vulnerable to hacking. All data stored in the server, especially password data needs to be encrypted to secure the data [1]. This is occurred because when there are a login and register process, calling and entering a username and password have some gaps that hackers can use it as the enter to the database as well as manipulating sourcecode (SQL Injection) and Sniffing.

In securing data, it has been developed such various way, including applying cryptography [2]. Cryptography is the art and science of maintaining the confidentiality of data where the original data is converted into other forms that cannot be read. Cryptographic algorithms used in securing passwords usually use MD5 encryption. However, using MD5 as a wider range of encryption techniques is very risky because even if you have to go through several steps to break into it, it can still be easily broken.

There are several cryptographic algorithms that are used to secure data, one of them is the AES algorithm. AES is a symmetry key encryption standard which was originally published with the Rijndael algorithm. The AES algorithm is a cryptographic algorithm besides being easy to implement, it is also quite reliable to date [3].

The problem of this research is how the analysis results of the security level from the login system on an android application by embedding the AES algorithm. The aim which to be achieved in this research is analyzing the level of security of the login system in an android application using the AES algorithm in variations of 3 keys 128, 192, and 256 bits.

## 2. METHOD

### 2.1. Cryptography

Cryptography is the study of mathematical techniques related to information security aspects such as confidentiality, data integrity, and authentication [4]. Cryptography is the science of encryption techniques which data encrypted using an encryption key to be something that is difficult to read by someone who does not have a decryption key. The encryption process is conducted using an algorithm with several parameters. In the cryptographic process there must be four main elements to run the cryptographic run well, which are most related to each other, namely: plain text, cipher text, cryptography key, and encryption decryption algorithm [5].

### 2.2. AES Algorithm

This AES algorithm was created with the aim to replace the DES algorithm that has long been used in encoding electronic data. After going through several selection stages, the Rijndael algorithm was specified as the AES cryptographic algorithm in 2000 [6]. In this algorithm there is an S-box, it is used to randomize input bits that

will result in output bits [7]. Substitution box (S-box) is a critical part of the data encryption and decryption procedures. The primary function of the S-box in advanced encryption standard algorithm is to randomize the 8-bit input into 8-bit output [8].

The AES algorithm uses substitution and permutation, and a number of rounds (repeated ciphers), which each round uses a different key (the key of each round is called a round key) [9]. AES sets the key lengths of 128, 192 and 256 bits. Therefore, it known as AES-128, AES-192, and AES-256. The summarizes of differences between the three AES versions shown in Table 1.

Table 1. AES version [8]

| Version | Number of Key (Nk word) | Block Size (Nb word) | Number of Round (Nr) |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

The sequence of data that has been formed in a 128-bit group is called a data block or text that will be encrypted to become a ciphertext. The AES key standard consists of 3 cipher blocks, namely AES-128, AES-192 and AES-256, which were adopted from a larger collection which was originally published as Rijndael [10].

The difference in key length will affect the number of rounds that will be implemented in this AES algorithm [11]. There are 10, 12, or 14 rounds in AES that match the size of the key used. Each round contains: replacement of bytes that are the same as DES, transition = line exchanges, mixed paths = left transition and XOR bits, Add sub-key = XOR key parts with cycle decisions.

AES has four main processes, namely AddRoundKey, which is a function that combines existing text ciphers with cipher keys, using XOR crosses [12]. SubBytes, exchanges the contents of an existing matrix or table with another matrix called Rijndael S-Box. ShifRows, is a process that performs shifts or shifts on each block / table element that is done per line. As in Figure 1, it shows the stages of the encryption process and the description of the AES algorithm.

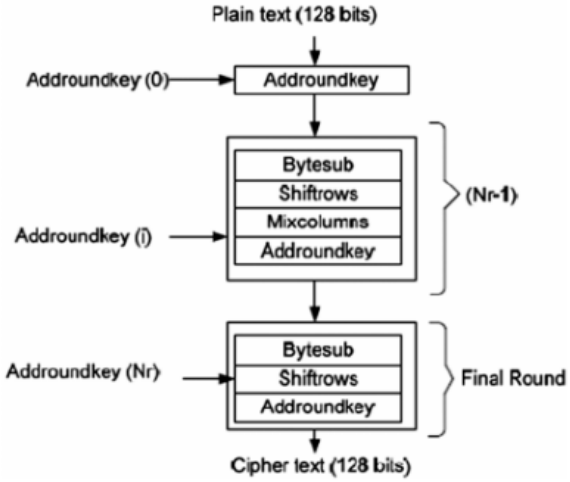


Figure 1. Encryption algorithm structure of AES

### 2.3. Mobile Application

The mobile application is a software that created and intended for portable smartphone devices which require the process of downloading mobile application software in the application store so that it can be used [13]. As for the type of store the application also varies like the Apple App Store, Play Store, or Blackbarry App.

Mobile applications have faster performance compared to mobile web [14]. Because in a mobile application only has one domain only and it's far more attractive in terms of visuals. For users, it also has full access to mobile devices this application. While the security and quality of this mobile application is far more guaranteed because it is controlled by the respective vendors.

### 2.4. Login System

According to Johnston [15], the login system (login, also called log in, log on, sign in, sign on, signin, sign in) is a process to access the computer by entering the identity of the user account and password to get the rights access using destination computer resources.

When logging in to enter the system, users will be asked to enter user identities such as user id and password in anticipation of system security. The password can be changed according the needs meanwhile the user id is never changed because it is a unique identity that refers to a particular user. If the two safeguards are successful or fulfilled, the user has the right to access the system [15].

The login process has a mechanism that consists of three stages, namely:

1. Identification. The stage which the user notifies their identity.
2. Authentication. The stage which the user verifies the user's claim is something known, such as a PIN code or password; something you have, such as magnetic cards; and something that becomes identity, such as fingerprint.

3. Authorization. The last stage, if the user identification is successful or correct, the system completes the login process and associates the user identity and access control information with the user session.

### 2.5. Authentication

One of the validation processes conducted during the login process is Authentication. When entering the system, the user's password is checked through a process that checks directly into the list to enter the system. This authorization is set up by the administrator, webmaster or site owner. Authentication is conducted so that the recipient of information can ensure the authenticity of the message comes from the person being asked for information [16]. In other meaning, the information that will be exist into system is really from the person who had authority. The authentication process in principle had a function as an opportunity for users and service providers in the process of accessing resources.

## 3. DISCUSSION

In this research, the experimental method was conducted to determine the level of system login security using the AES algorithm using 3 key variations of 128, 192, and 256 bits. The method implementation used by designing an android-based of system login application. This system login was developed using the Android Studio Framework version 3.0.1 with JAVA Script, PHP and MYSQL databases.

In the AES encryption calculation process, there are two objects that will be processed, namely the plaintext and key, which will produce a ciphertext (encrypted text). The first process is to change the plaintext and key to hexadecimal.

In the implementation, the AES algorithm has three key length variations, each of them has a key length of 128-bit, 192-bit, and 256-bit in each character that means 8 bits long. The round of each key is different where 128-bit have 10 rounds, 192-bit have 12 rounds, and 256-bit have 14 rounds. In the AES process it also has 2 fixed variables namely Rcon and S-Box in encryption and decryption.

Roundkeys (cipherkey) and ciphertext are two important outputs in the AES encryption process, which roundkeys is the initial output that must be known to get ciphertext. For looking for the roundkeys, it must go through several stages such as: subbytes, shift rows, and mix columns. This stage will be repeated 10 times. As for the ciphertext the steps are: subbytes, shift row, mix column and addroundkey. This stage was conducted in 10 rounds. For other key lengths such as 192-bit, 12 rounds are performed while 256 -bit are conducted in 14 rounds.

The initial process in AES encryption is transforming the plaintext and key into hexadecimal form. In Figure 2, the text password is shown which will be encrypted when registering by entering the key.

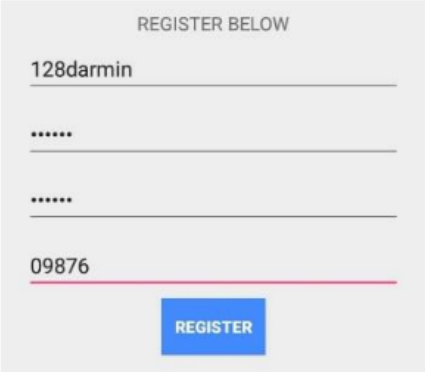



Figure 2. The interface of sending personal data when registering to database

The key in hexadecimal form is used as a 4x4 matrix so that it gets w [0], w [1], w [2] and w [3]. It will go through the shiftrow process is the last 4 bytes w [3] will be shifted by 1 byte, so that when the initial sequence formation is 0,1,2,3 to 1,2,3,0. After passing through, the process will enter the next process, namely, substitution. The acquisition value of each byte substituted with AES S-Box will become a new matrix. This matrix which will be multiplied by the Rcon table, so that it gets the value g (w [3]). The multiplication result which means g (w [3]) will be conducted using XOR operation with the initial 4 bytes (w [0]) which will get a new matrix line, w [4].

For getting a new matrix w [5], w [6], or w [7] can be conducted by the formula:

$$W[i] = w[i-1] \text{ XOR } w[i-4] \quad (1)$$

Then, look for w [8] is conduct the same thing when looking for w [4]. Whereas for w [9], w [10], w [11] can be calculated using the formula above. Likewise, so on when multiples of 4, the first one is searched by way of w [4] and is conducted as many as 10 rounds.

After getting all the roundkeys, the next step is processing the plaintext that is already in hexadecimal form to ciphertext. By doing the ten rounds process, you will get the final result in the form of ciphertext. Each round must go through a sequential process stage, from addrounkeys, subtitution, shiftrow and mix column. These stages will be repeated 10 times.

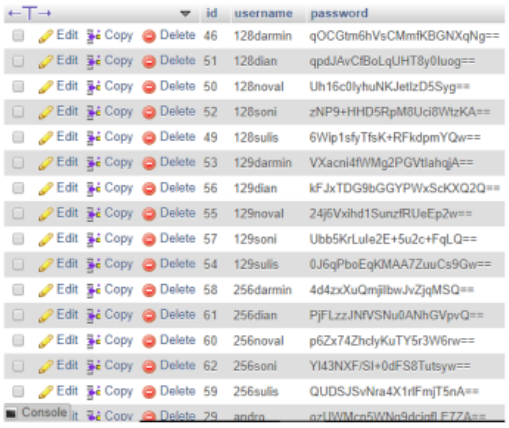
The first stage is roundkeys. The roundkey that has been obtained will be conducted using XOR with a plaintext that is already in ASCII form and has been transformed into a matrix. In conducting of XOR, each byte in each matrix is made a binner form so that it is easy to calculate. After XOR is complete, a new matrix will be obtained. The matrix will then enter in the second stage, namely substitution. At this stage the new matrix will be substituted with the AES S-Box.

The substituting stage is conducted by taking for each byte in the new matrix which in each case consists of two characters which the first character refers to the line, while

the second character refers to the column. This cross lining between rows and columns will form a new matrix again. The substitution result matrix will then enter in the 3rd process, namely shiftrow. This step will shift to each byte in each row in the order of shifting the shift is 0,1,2,3 in each row. After the shift process is done, it will get a new matrix sequence. Next this matrix will enter in the mix-column process which is multiplying the matrix with Rcon. After the process is complete, a new matrix will be obtained and will be the first round.

From the addrounkeys stage to the mix-column, it will be repeated 10 rounds. The results of the 10th round or the last round will later be ciphertext. The process applies to 128-bit key variations, while for 192 bits the process rotation reaches 12 turns to get ciphertext and at 256 bits it reaches 14 rounds to get ciphertext. Where the results of this encryption in the form of ciphertext will be stored in the database. For more details, the display of AES encryption stored in the database can be seen in Figure 3.

In addition to the AES encryption process that converts the plaintext into a ciphertext and an AES decryption process which is the opposite of the encryption process whose function is to convert the ciphertext to its original form, namely plaintext. In the decryption process it is almost the same as the encryption process except that when encryption is used it is plaintext while decryption is ciphertext with key as a bridge for each process. The form of decrypted ciphertext by entering the key shown in Figure 4.



| | id | username | password |
|---|---|---|---|
| Edit Copy Delete | 46 | 128darmin | qOCGtm6hVsCMmfKBGNXqNg== |
| Edit Copy Delete | 51 | 128dian | qpdJAvCfBoLqUHT8y0luog== |
| Edit Copy Delete | 50 | 128noval | Uh16c0lyhuNKJetlzD5Syg== |
| Edit Copy Delete | 52 | 128soni | zNP9+HHD5RpM8Uci8WtzKA== |
| Edit Copy Delete | 49 | 128sulis | 6Wlp1sfyTfsK+RFkdpmYQw== |
| Edit Copy Delete | 53 | 129darmin | VXacni4fWMg2PGVtlahqjA== |
| Edit Copy Delete | 56 | 129dian | kFJxTDG9bGGYPWxScKXQ2Q== |
| Edit Copy Delete | 55 | 129noval | 24j6Vxihd1SunzfRUeEp2w== |
| Edit Copy Delete | 57 | 129soni | Ubb5KrLule2E+5u2c+FqLQ== |
| Edit Copy Delete | 54 | 129sulis | 0J6qPboEqKMAA7ZuuCs9Gw== |
| Edit Copy Delete | 58 | 256darmin | 4d4zxXuQmjllbwJvZjqMSQ== |
| Edit Copy Delete | 61 | 256dian | PjFLzzJNfVSNu0ANhGVpvQ== |
| Edit Copy Delete | 60 | 256noval | p6Zx74ZhclyKuTY5r3W6rw== |
| Edit Copy Delete | 62 | 256soni | Yl43NXF/SI+0dFS8Tutsyw== |
| Edit Copy Delete | 59 | 256sulis | QUDSJSvNra4X1rlFmjT5nA== |

Figure 3. The interface of aes encryption in database


Figure 4. The interface of password ciphertext in decryption

## 4. RESULT

The system testing was conducted by using a trial of several samples. There are 5 samples at each length of the AES key. The testing result used of sample user passwords when the registry can be seen in Table 2.

Table 2. The testing result of AES encyrption

| | Username | Password | Key | Ciphertext |
|---|---|---|---|---|
| 128 *bit* | 128darmin | qwerty | 09876 | qOCGtm6hVsCMmfKBGNXqNg== |
| | 128sulis | qwertyu | 098765 | 6Wip1sfyTfsK+RFkdpmYQw== |
| | 128noval | qwertyui | 0987654 | Uh16c0IyhuNKJetIzD5Syg== |
| | 128dian | qwertyuio | 09876543 | qpdJAvCfBoLqUHT8y0Iuog== |
| | 128soni | qwertyuiop | 098765432 | zNP9+HHD5RpM8Uci8WtzKA== |
| 192 *bit* | 192darmin | qwerty | 09876 | VXacni4fWMg2PGVtIahqjA== |
| | 192sulis | qwertyu | 098765 | 0J6qPboEqKMAA7ZuuCs9Gw== |
| | 192noval | qwertyui | 0987654 | 24j6Vxihd1SunzfRUeEp2w== |
| | 192dian | qwertyuio | 09876543 | kFJxTDG9bGGYPWxScKXQ2Q== |
| | 192soni | qwertyuiop | 098765432 | Ubb5KrLuIe2E+5u2c+FqLQ== |
| 256 *bit* | 256darmin | qwerty | 09876 | 4d4zxXuQmjiIbwJvZjqMSQ== |
| | 256sulis | qwertyu | 098765 | QUDSJSvNra4X1rlFmjT5nA== |
| | 256noval | qwertyui | 0987654 | p6Zx74ZhclyKuTY5r3W6rw== |
| | 256dian | qwertyuio | 09876543 | PjFLzzJNfVSNu0ANhGVpvQ== |
| | 256soni | qwertyuiop | 098765432 | Yl43NXF/SI+0dFS8Tutsyw== |

In the experiment AES encryption above, it used the same password and key, which is applied to each key length 128, 192, and 256 bit with different lengths of each password and key in each AES category. The encryption experiment showed that using the input of the same password and key which processed in each long version of the AES key it produced different ciphertexts. It was occured because the processing on each long version of the AES key was different from each other.

Another result in the implementation of the AES algorithm in each key variation was its ciphertext that had decrypted would be the same as before when entering the same key while the encryption process. If in inputting ciphertext data or key was wrong then the decryption result would not come out or failed.

The next testing was about of the attacks on the system. The attack used SQL Injection on systems that have not been installed by the AES algorithm and the system that has been installed by the AES algorithm. The results of experiments on systems that had not installed the AES algorithm can be seen in Table 3.

Table 3. The attacks of SQL injection in a login system without and using AES

| No. | SQL Injection | Without AES | Using AES | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Attack Impact | Attack Impact | | | | | |
| | | | 128 bit | | 192 bit | | 256 bit | |
| | | | T | F | T | F | T | F |
| 1 | 'or 1=1# | Successfully Login | - | √ | - | √ | - | √ |
| 2 | “or 1=1- | Failed to Login | - | √ | - | √ | - | √ |
| 3 | „or 1=1- | Failed to Login | - | √ | - | √ | - | √ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | „or”a”=”a”# | Failed to Login | - | √ | - | √ | - | √ |
| 5 | 'or"a"="a"# | Successfully Login | - | √ | - | √ | - | √ |
| 6 | Admin” # | Failed to Login | - | √ | - | √ | - | √ |
| 7 | ' or 'x'='x'# | Successfully Login | - | √ | - | √ | - | √ |
| 8 | Admin” or 1=1 # | Failed to Login | - | √ | - | √ | - | √ |
| 9 | hi' or 'a'='a'# | Successfully Login | - | √ | - | √ | - | √ |
| 10 | hi”) or (“a”=”a | Failed to Login | - | √ | - | √ | - | √ |
| 11 | Admin' or 0=0# | Successfully Login | - | √ | - | √ | - | √ |
| 12 | “or ”a”=”a | Failed to Login | - | √ | - | √ | - | √ |
| 13 | admin' OR '1'='1 | Successfully Login | - | √ | - | √ | - | √ |
| 14 | ' or 0=0 # | Successfully Login | - | √ | - | √ | - | √ |
| 15 | ' or 0=0 -- | Failed to Login | - | √ | - | √ | - | √ |
| 16 | " or 0=0 -- | Failed to Login | - | √ | - | √ | - | √ |
| 17 | " or 0=0 # | Failed to Login | - | √ | - | √ | - | √ |
| 18 | admin ' or 'x'='x | Successfully Login | - | √ | - | √ | - | √ |
| 19 | " or "x"="x | Failed to Login | - | √ | - | √ | - | √ |
| 20 | „) or ('x'='x | Failed to Login | - | √ | - | √ | - | √ |
| 21 | ' or 1=1-- | Failed to Login | - | √ | - | √ | - | √ |
| 22 | " or 1=1-- | Failed to Login | - | √ | - | √ | - | √ |
| 23 | or 1=1-- | Failed to Login | - | √ | - | √ | - | √ |
| 24 | ' or a=a-- | Failed to Login | - | √ | - | √ | - | √ |
| 25 | " or "a"=”a | Failed to Login | - | √ | - | √ | - | √ |
| 26 | hi” or 1=1 – | Failed to Login | - | √ | - | √ | - | √ |
| 27 | admin”– | Failed to Login | - | √ | - | √ | - | √ |
| 28 | ”having 1=1– | Failed to Login | - | √ | - | √ | - | √ |
| 29 | hi' or 'a'='a | Failed to Login | - | √ | - | √ | - | √ |
| 30 | hi" or 1=1 -- | Failed to Login | - | √ | - | √ | - | √ |
| 31 | “or 0=0 – | Failed to Login | - | √ | - | √ | - | √ |
| 32 | admin 'or"a"="a"# | Successfully Login | - | √ | - | √ | - | √ |
| 33 | hi" or "a"="a | Failed to Login | - | √ | - | √ | - | √ |
| 34 | hi' or 1=1 -- | Failed to Login | - | √ | - | √ | - | √ |
| 35 | hi' or 'a'='a# | Failed to Login | - | √ | - | √ | - | √ |
| 36 | hi') or ('a'='a | Failed to Login | - | √ | - | √ | - | √ |
| 37 | hi") or ("a"="a# | Failed to Login | - | √ | - | √ | - | √ |
| 38 | admin hi' or | Failed to Login | - | √ | - | √ | - | √ |
| 39 | a'='a'# | Successfully Login | - | √ | - | √ | - | √ |
| 40 | admin'or 1=1# | Successfully Login | - | √ | - | √ | - | √ |
| | admin hi' or 'a'='a'# | | | | | | | |

Information: T = true (Successfully Login)
F = false (Failed to Login)

From the forty SQL Injection samples there were eleven who could log into the system (27.5%) and the rest failed to enter (72.5%). It indicates that a normal system login without security algorithms will be vulnerable to attack, so that illegal users can easily entered the system and can steal data contained within it.

From the forty attack techniques using SQL Injection nothing could be entered or failed to log in to the system which obtained (100%). This result was also obtained at the other AES key lengths such as 192 bits and 256 bits with a percentage of 100%. This proved that in all variations of AES key length could secure data from SQL

Injection attacks. The execution time needed for the encryption and decryption process in each variation of the AES key using the same key length and plaintext can be seen in Table 4.

Table 4. Average time of encryption and decryption

| | Average time of encryption (second) | Average time of decryption (second) |
|---|---|---|
| AES 128 | 5,8 | 3,54 |
| AES 192 | 7,74 | 5,8 |
| AES 256 | 9,46 | 8,26 |

In Table 5, it can be seen that 128 bit AES required an average time of 5.8 for encryption and 3.54 for decryption of passwords, whereas AES 192 requires an average time of 7.74 for encryption and 5.8 for decryption, for AES 256 requires an average time of 9.46 for encryption and 8.26 for decryption. It showed that the longer the key variation in AES takes a long time to do the encryption process. This is occured because the AES 256 encryption process is quite long compared to the key variations below, so the decryption process will be even longer. So, it will take a long time for hackers to find the key, this proves AES 256 for better security than the key variations below.

## 5. CONCLUSION

The AES algorithm is able to secure the personal data very well on 3 keys variations 128-bit, 192-bit and 256-bit. It was indicated by the password as the plaintext can be encrypted properly on each key variation of AES, which the key is only known by user because it is not entered into the database. The security level of testing is also conducted using the attack of 40 SQL Injection samples where system logins without security was obtained percentage of 27.5% can enter the system which it compared to login systems that used AES algorithms 128-bit, 192-bit or 256-bit was obtained percentage of 100% that cannot enter into the system. The estimation of the average encryption process AES 128-bit, 192-bit and 256-bit are 5.8 seconds, 7.74 seconds, and 9.46 seconds. Thus, the level of security in each AES key indicated that the key with the highest bit length will be more difficult to hack because it requires a considerable amount of time compared to the smaller key lengths.

## 6. REFERENCES


[1] Alamsyah. (2013). Membangun Sistem Pemilu Online Menggunakan Advanced Encryption Standard (AES). *UNNES Journal of Mathematics*, 2(2), 133-142.

[2] Prasetiyo, B., Gernowo, R., & Noranita, B. (2014). Kombinasi Steganografi Berbasis Bit Matching dan Kriptografi DES untuk Pengamanan Data. *Scientific Journal of Informatics, 1*(1), 79-94.

[3] Putra, F., Budiman, G., & Andini, N. (2015). Perbandingan dan Analisis Performansi Enkripsi-Dekripsi Teks Menggunakan Algoritma AES dan AES yang Termodifikasi Berbasis Android. *e-Proceeding of Engineering, 2*(2), 3022-3030.

[4] Menezes, P., Oorschot, V., Vanstone, S. (1996). *Handbook of Applied Cryptography*. CRC Press: USA.

[5] Rahardjo, B. (2003). *Memahami Model Enkripsi & Security Data*. Yogyakarta: Penerbit Wahana Komputer dan Andi Offset.
[6] Gladman, B. (2003). A Specification for Rijndael, The AES Algorithm. *Journal Springer-Verlag*.
[7] Alamsyah. (2017). AES S-Box Construction Using Different Irreducible Polynomial and Constant 8-bit Vector. IEEE Conference on Dependable and Secure Computing, 366-369.
[8] Alamsyah, Agus, B., & Adji, T.B. (2018). The Replacement of Irreducible Polynomial And Affine Mapping For The Construction Of A Strong S-Box. Springer Science + Business Media B.V., part of Springer Nature 2018, 93(4).
[9] Munir, R. (2006). *Kriptografi*. Bandung: Informatika Bandung.
[10] Tampubolon, N.B., Isnanto, R.R, & Sinuraya, E.W. (2015). Implementasi Dan Analisis Algoritma Advanced Encryption Standard (AES) Pada Tiga Variasi Panjang Kunci Untuk Berkas Multimedia. *Jurnal TRANSIENT Universitas Diponegoro*, *4*(4), 1008-1012.
[11] Primartha, R. (2013). Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Advanced Encryption Standard (AES). *Journal of Research in Computer Science and Applications, 2*(1), 13-23.
[12] Shahbazi, K., Eshghi, M., & Mirzaee, R.F. (2017). Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5. *Engineering Science and Technology, an International Journal,* (20), 1308-1317.
[13] Nurhadryani, Y., Sianturi, S.K., Hermadi, I., & Khotimah, H. (2013). Pengujian Usability untuk Meningkatkan Antarmuka Aplikasi Mobile. *Jurnal Ilmu Komputer Agri-Informatika, 2*(2), 83-93.
[14] Utomo, E. P. (2013). *Mobile Web Programming*. Yogyakarta: Penerbit Andi.
[15] Khairina, D.M. (2011). Analisis Keamanan Sistem Login. *Jurnal Informatika Mulawarman,* 6(2), 64-76.
[16] Santoso, K.I., Sediyonob, K., & Suhartono. (2013). Studi Pengamanan Login Pada Sistem Informasi Akademik Menggunakan Otentifikasi One Time Password Berbasisis SMS dengan Hash MD5. *Jurnal Sistem Informasi Bisnis, 1*(1), 7-12.

ORIGINALITY REPORT

| 17% | 13% | 11% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | e-journal.uajy.ac.id<br>Internet Source | 4% |
| 2 | documents.iss.net<br>Internet Source | 2% |
| 3 | docplayer.info<br>Internet Source | 1% |
| 4 | pdfs.semanticscholar.org<br>Internet Source | 1% |
| 5 | www.coursehero.com<br>Internet Source | 1% |
| 6 | Rismayani, Cucut Susanto. "Using AES and DES Cryptography for System Development File Submission Security Mobile-Based", 2020 8th International Conference on Cyber and IT Service Management (CITSM), 2020<br>Publication | 1% |
| 7 | staff.unnes.ac.id<br>Internet Source | 1% |
| 8 | Hesti Putri Winasih, Eko Hari Rachmawanto, Christy Atika Sari, De Rosal Ignatius Moses Setiadi. "Implementation of LSB-RSA | 1% |

Algorithm for the Authenticity of the JPG File Certificate", 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), 2020
Publication

| | | |
|---|---|---|
| 9 | Henny Indriyawati, Titin Winarti, Vensy Vydia. "Web-based document certification system with advanced encryption standard digital signature", Indonesian Journal of Electrical Engineering and Computer Science, 2021<br>Publication | 1 % |
| 10 | Apri Siswanto, Yudhi Arta, Evizal Abdul Kadir, Bimantara. "Chapter 20 Text File Protection Using Least Significant Bit (LSB) Steganography and Rijndael Algorithm", Springer Science and Business Media LLC, 2021<br>Publication | 1 % |
| 11 | gopellive.blogspot.com<br>Internet Source | 1 % |
| 12 | aircconline.com<br>Internet Source | <1 % |
| 13 | Nurhayati, Syukri Sayyid Ahmad. "Steganography for inserting message on digital image using least significant bit and AES cryptographic algorithm", 2016 4th International Conference on Cyber and IT Service Management, 2016<br>Publication | <1 % |

| | | |
|---|---|---|
| 14 | docplayer.net<br>Internet Source | <1 % |
| 15 | iml.ece.mcgill.ca<br>Internet Source | <1 % |
| 16 | interview2me.blogspot.com<br>Internet Source | <1 % |
| 17 | www.absoluteastronomy.com<br>Internet Source | <1 % |
| 18 | Alamsyah, Agus Bejo, Teguh Bharata Adji. "The replacement of irreducible polynomial and affine mapping for the construction of a strong S-box", Nonlinear Dynamics, 2018<br>Publication | <1 % |
| 19 | Ria Andriani, Stevi Ema Wijayanti, Ferry Wahyu Wibowo. "Comparision Of AES 128, 192 And 256 Bit Algorithm For Encryption And Description File", 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE), 2018<br>Publication | <1 % |
| 20 | Alamsyah, Agus Bejo, Teguh Bharata Adji. "AES S-box construction using different irreducible polynomial and constant 8-bit vector", 2017 IEEE Conference on Dependable and Secure Computing, 2017<br>Publication | <1 % |
| 21 | www.toptenreviews.com<br>Internet Source | <1 % |

| | | |
|---|---|---|
| 22 | F H S Pratama. "Designing an online-based questionnaire application for mobile devices", Journal of Physics: Conference Series, 2020<br>Publication | <1 % |
| 23 | www.tandfonline.com<br>Internet Source | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 10 words |
| Exclude bibliography | On | | |

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11